
ENERO 1987
REVISTA PARA USUARIOS DE LA NORMA
LOAD MSX
Año 1 Nº 9
₳ 2,70 Rep. Argentina
MUSICALIZADORES
PROGRAMAS INEDITOS PARA TALENT, TOSHIBA Y SVI
COMO ORDENAR DATOS
APLICACIONES PARA TOSHIBA
NUEVO CONCURSO



A la Talent MSX nada le es imposible

El Club de Usuarios de MSX ya funciona en sus tres direcciones: CABILDO 2027 - 1º A CORDOBA 654 - P.B. (1054) y TUCUMAN 2044 - 1º - CAPITAL

Invitamos a los felices usuarios de la TALENT MSX al **curso gratuito** de introducción al fabuloso mundo de MSX.

Participe del Club de Usuarios de MSX y encuéntrese con sus amigos que también tienen la TALENT MSX, e intercambiará programas, datos y chimentos.

Podrá probar todos los accesorios de la línea MSX, ¡¡desde disketteras hasta robots!!

Podrá ver y leer todo lo que le interese sobre la norma MSX: catálogos, libros y revistas de todo el mundo. Todo con la seguridad, respaldo y seriedad que sólo TALENT puede brindarle.

¡Para inscribirse, no olvide traer su factura de compra!

Club Talent MSX

LOAD MSX

Director General
Ernesto del Castillo

Director Editorial
Cristian Pusso

Director Periodístico
Fernando Flores

Director Financiero
Javier Campos Malbrán

Secretario de Redacción
Ariel Testori

Redacción
Andrea Sabin Paz

Arte y Diagramación
Fernando Amengual y Tamara Migelson

Departamento de Avisos
Oscar Devoto y Nelso Capello

Departamento de Publicidad
Guillermo González Aldalur

Servicios Fotográficos
Image Bank, Oscar Burriel, Víctor Grubicy y Eduardo Comesaña

Load Revista para usuarios de la norma MSX es una publicación mensual editada por Editorial PROEDI S.A., Paraná 720, 5° Piso, (1017) Buenos Aires. Tel.: 46-2886 y 49-7130. Radiollamada: Tel.: 311-0056 y 312-6383, código 5941. Registro Nacional de la Propiedad Intelectual: E. T. M. Registrada. Queda hecho el depósito que indica la Ley 11.723 de la Propiedad Intelectual.

Precio de este ejemplar: A 2,70
Impresión: Calcotam, Fotocromo tapa: Columbia. Fotocomposición: Interamericana Gráfica.
Los ejemplares atrasados se venderán al precio del último número en circulación.

Distribuidor en Capital: Martino, Juan de Garay 358, P. B. Capital.
Distribuidor interior: D G P: Hipólito Yrigoyen 1450, Capital Federal. T.E. 38-9266/9800.

CORREO ARGENTINO SUC. CENTRAL — FRANQUEO PAGO CONCESION Nº 2538 — TARIFA REDUCIDA

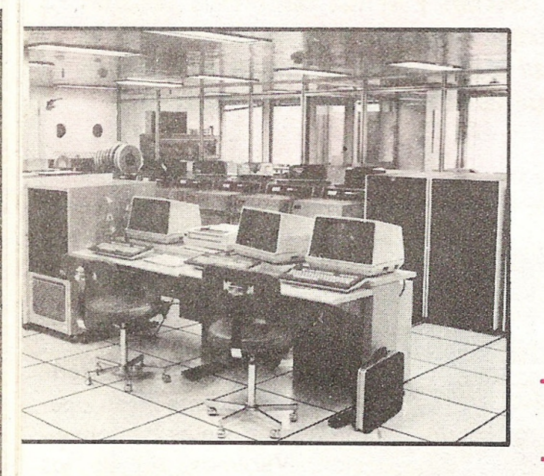
"NOS DESPIERTA LA MENTE"

Este comentario lo hizo el ganador del concurso organizado por nuestra revista, al referirse a este certamen, y agregó que de esta manera no se "deja caer a la MSX como una simple máquina de juegos". Precisamente ése es nuestro objetivo: alentar al desarrollo de la inteligencia.
Víctor Villarraza nos dijo que todo lo que sabe lo aprendió en su casa, haciendo programas de prueba y practicando rutinas ("es la mejor forma de aprender" recalcó). Y además compartió información, trucos y secretos con usuarios de su ciudad, Paraná. "La computadora nuclea a muchos y ayuda también a ser más sociable", enfatizó al respecto. Y justamente ésa es la intención de *Load MSX*: reunir a amigos que juntos transitan el apasionante mundo de la informática.

LOS EDITORES

CARTRIDGE BANCARIO

En el marco del convenio que Telemática ha firmado con el Banco del Buen Ayre, la productora de la Talent MSX desarrolló un nuevo medio para que los clientes de esa entidad crediticia cuenten con los servicios de la banca electrónica. Se trata de un cartucho que contiene todas las instrucciones para poder acceder a la red. Para lograrlo antes sólo se podía a través de diskettes o cassettes. En el primer caso, el costo de la diskettera, y en el segundo el tiempo que implicaba cargar el programa, eran impedimentos para el fácil acceso al sistema, algo que con los cartuchos se soluciona, ya que son de bajo costo y rapidez operativa.



Cabe recordar que el sistema de banca electrónica del Banco del Buen Ayre incluye la posibilidad de realizar consultas de saldos, transferencias de fondos, compras por catálogo, listados de movimientos realizados y otros servicios. El software fue desarrollado, en su totalidad, por Telemática.

GAME CREATOR

Las tareas de Microsoft en MSX no se agotan. Han creado un programa titulado GAME CREATOR, es decir, creador de juegos, que se lanzará en cassettes a bajo costo. Este soft especial permite que hasta el más inexperto programador cree sus propios juegos, inclusive sprites, música y secuencias animadas. Todo se puede controlar con un simple joystick.

El único consejo que propone Microsoft antes de utilizar GAME CREATOR, es practicar jugando con varios de los programas que se adquieren en los comercios y luego sí, empezar a crear los propios.

LENGUAJES PARA TALENT MSX

La Talent MSX puede operar con diversos lenguajes. Entre ellos, destacan los técnicos de la firma Telemática, los que actualmente se encuentran disponibles en el país son: Macro-Assembler con Linkeditor; MSX C; Turbo Pascal; Microprolog (intérprete); Lisp (editor); Logo; Basic (compillador); RM Cobol (Compilador); Cobol 80 (comp.) y D Base II.

COMPUTADORAS EN LA PLAYA

Desde fines de diciembre funcionan en Pinamar cuatro centros de información turística para brindar a quienes visitan el balneario datos actualizados sobre disponibilidades de alojamiento hotelero, estado y pronóstico del tiempo, acontecimientos culturales, deportivos y sociales y nómina de comercios, entre otros datos útiles.
Lo característico de dichos centros es que están equipados con sistemas Talent MSX, cada uno de los cuales está conformado por un teclado, monitor, impresora y diskettera. Telemática formó al personal que opera las computadoras, que depende de la Municipalidad local, y también desarrolló el programa de operación del sistema.

CONVENIO Y CONCURSO

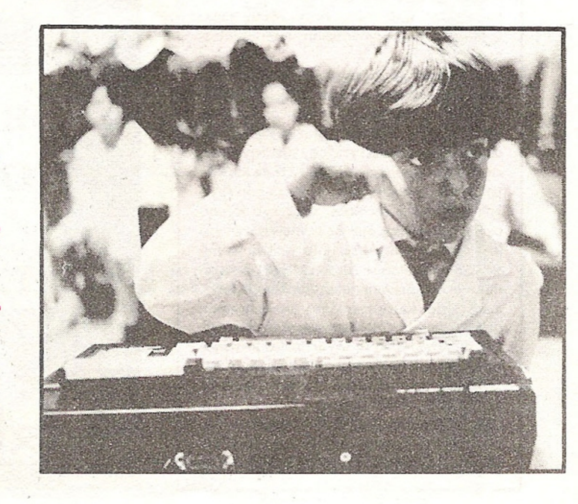
Por medio de un convenio realizado entre la empresa Telemática y el IDES (Instituto de Enseñanza Superior) este último llama a concurso para el otorgamiento de licencias para la representación del mismo en la Capital Federal y el Gran Buenos Aires.
Estos futuros licenciatarios podrán desarrollar diversas actividades en zonas designadas con exclusividad, ofreciendo una amplia gama de servicios a la comunidad. Entre ellos encontramos: Dictado de cursos para usuarios de microcomputadoras y para profesionales interesados en las posibilidades de este tipo de herramienta, carreras de capacitación, creación de un centro zonal de usuarios donde se pueden evacuar diversas preguntas, intercambiar y comentar experiencias y hasta consultar bancos de datos, entre otros.

Así Telemática brinda todo su apoyo pedagógico y asesoramiento permanente en todas las áreas que abarca este proyecto.

ARCHIVOS EN LOGO

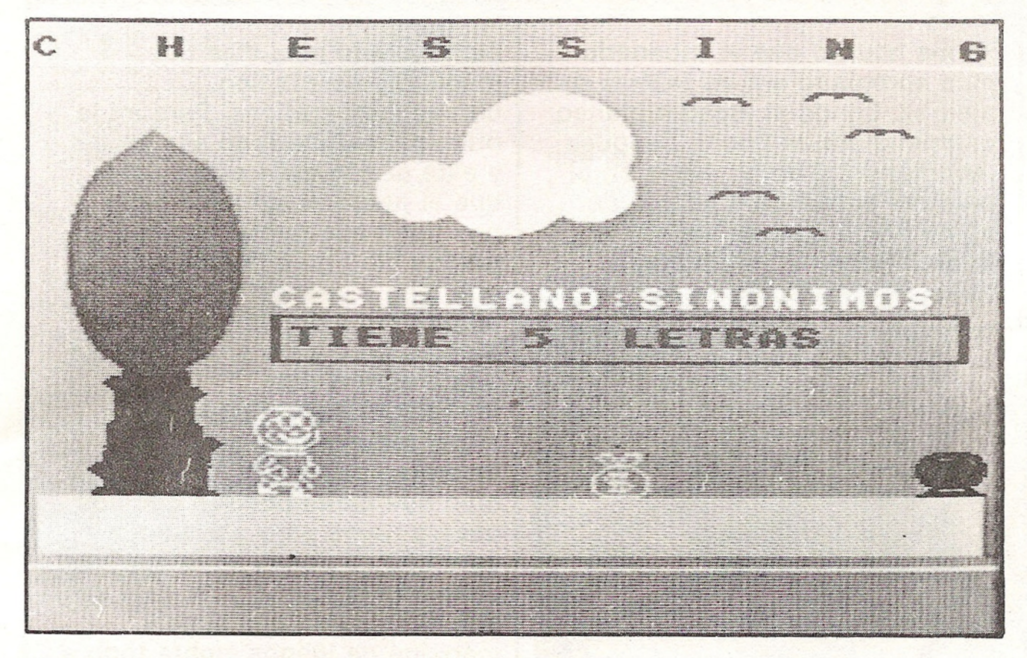
La posibilidad de trabajar con archivos con el lenguaje Logo en MSX es ya una realidad gracias al software que han desarrollado para las Talent. En forma sintética, son una serie de "primitivas" nuevas que se incluyen dentro del Logo, las que permiten el uso de archivo, cosa que hasta ahora no podía realizarse en ese lenguaje.

El diskette que contiene este soft todavía no se puede adquirir a nivel comercial, porque aún no se ha hecho el manual de instrucciones, pero pronto estará disponible para el público.



LOS MEJORES PROGRAMAS PARA MSX

Victor Villarraza ganó el certamen organizado por Load MSX y auspiciado por Telemática, por un original juego de inteligencia. El segundo premio lo obtuvo Daniel Francesch, con un utilitario muy completo.



La elección del jurado fue difícil debido al alto nivel de la mayoría de los participantes.

Se recibieron programas de gestión, educativos, entretenimientos originales, y otros de inteligencia artificial.

El ganador fue Víctor Villaraza quien obtuvo el periférico Talent y la beca provista por Telemática S.A. Este entrerriano creó un programa muy original, bajo el nombre de "CHESSING", que es una mezcla del estanciero, el ajedrez y los dados, más preguntas educativas.

El software, totalmente realizado en Basic, es sumamente entretenido. Continuamente el juego nos presenta distintos estados y variaciones que hacen mantener a la computadora y al jugador en constante comunicación. El autor conjugó reglas de conocidos juegos de mesa.

Pero no sólo basta con tomar decisiones y tener la suerte de nuestro lado para ganar, sino que se debe contar con conocimientos de gramática y geografía.

En síntesis, este amigo entrerriano demostró poder dominar la programación Basic de esta norma estándar.

Pero también hablemos del segundo triunfador que ganó un periférico Talent provisto por Telemática S.A. Fue un porteño de 18 años llamado Daniel Francesch. Ingresó al mundo de los ordenadores hace un par de años, pero demostró con su programa "GAFIC II", que no hace falta demasiado tiempo para crear excelentes programas.

Se trata de un software que ayuda al usuario a crear fantásticas pantallas.

Pero no se termina al formar efectos visuales multicolores. También podemos guardar nuestras creaciones para luego incorporarlas en otros programas.

Daniel nos contó que a este programa lo fue componiendo lentamente, agregándole de a poco las distintas opciones con las que ahora cuenta.

Este es un utilitario muy completo, pero no por eso difícil de manejar. El software incluye un menú con la explicación de las opciones.

Se puede manejar con joystick o teclado: dos posibilidades que Daniel tuvo en cuenta para los distintos usuarios.

Grafic II está formado por un programa principal en BASIC y dos rutinas en ASSEMBLER, demostrando el ganador un total dominio sobre una MSX.

A Víctor y a Daniel, nuestras felicitaciones por haber realizado excelentes programas.

Seleccionar las 3 menciones no resultó tarea sencilla.

La primera de ellas fue un programa llamado "Evolución" realizado por Nicolás Dario Moyano que muestra los comienzos del hombre, su evolución junto al planeta, marca los hitos más importantes de este siglo como la primera guerra mundial, la segunda, la aparición de las computadoras, y la llegada del hombre a la Luna.

Comienza con un paisaje desolado, porque después de tanta evolución tal vez el fin del planeta sea una realidad y comience todo de nuevo, con un paisaje similar.

Este soft está desarrollado en BASIC y Assembler con buena resolución gráfica. Pertenece a la categoría de educativos.

La segunda mención se llama "Mi República Argentina" y es obra de Pedro Favio Azcui, de Avellaneda, provincia de Buenos Aires. Es Educativo, y nos presenta un excelente paseo por toda la geografía de nuestro país. Pedro Favio nos dijo que le llevó mucho trabajo la generación de las pantallas de las provincias, y que lo fue haciendo de a poco.

Los datos de superficie con que cuenta el programa son del Instituto Geográfico Militar y todas las cifras de población fueron consultadas al Instituto Nacional de Estadísticas y Censos (del último censo del año 1980).

La última mención corresponde al programa titulado "El Corazón", donde se muestra precisamente su funcionamiento.

Lo ideó Carlos Alejandro Escobar, de San Pedro de Jujuy.

También es educativo.

Nos sorprendió muy gratamente ver la resolución de las pantallas y el profun-

do conocimiento que tiene sobre el tema.
Podremos ver y escuchar los latidos del corazón, desde el de un elefante hasta el de un pequeño picaflor. Es un viaje interesante por nuestro interior, cosa que habitualmente no resulta simple.
Finalmente agradecemos a todos los participantes y los invitamos a enviar trabajos al nuevo concurso anunciado por nuestra revista.

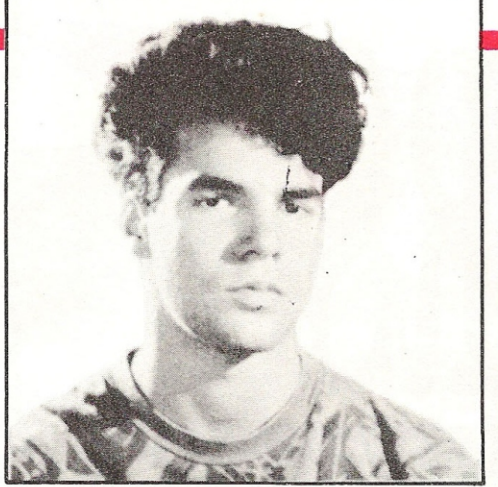

Daniel Francesch

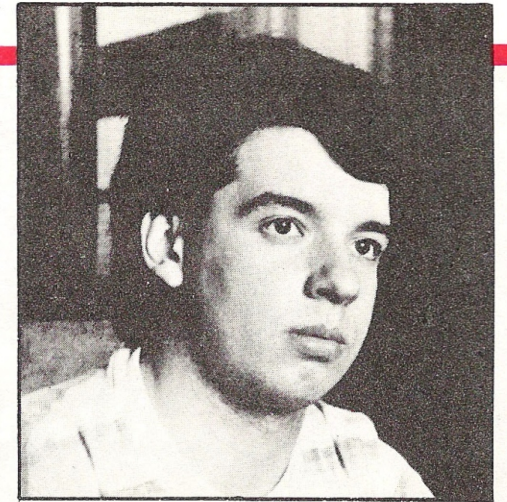
"LA MEJOR FORMA DE APRENDER"

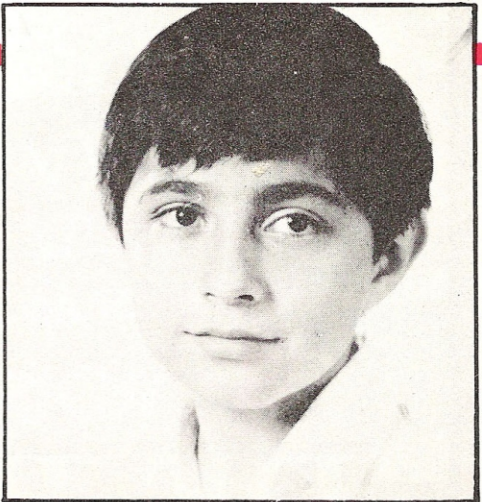
Realmente, me siento muy feliz de haber participado con éxito en este concurso. Estoy conforme con el trabajo, pero como ya dije, la MSX es muy nueva y la tarea de aprender a usarla, sobre todo en el interior, recién comienza. Por mi parte, todo lo que sé lo aprendí en mi casa, haciendo programas de prueba y practicando rutinas. Creo que es la mejor manera de aprender, porque para mí, aprender y comprender el BASIC es como aprender a hablar, hay que escuchar, repetir y practicar, aunque a veces salga mal.
CHESSING está todo en BASIC, ya que con el lenguaje de máquina no me llevo nada bien.
A mí me gusta mucho los juegos de mesa, y cuando me sugirieron la idea del "Estanciero" comencé a ambientar el mío. Tenía que ser algo con casilleros, y estaba leyendo una revista cuando vi el juego "Archon" para Commodore. Ahí se me ocurrió que el escenario más apropiado era un tablero de ajedrez. Donde el jugador blanco competía con el negro. Después recordando el libro "Alicia en el país del espejo", me acordé de cómo Alicia cuando llegaba al final de tablero, se coronaba reina. Fue así como se me ocurrió el desenlace de mi juego, una coronación. Pero tenía que introducir plata, y para esto programé que hubiera que comprar tierras. También le agregué tarjetas de la suerte con 8 variantes al azar. Con todo esto, todavía le faltaba algo al juego, por lo que decidí introducirle una pantalla adicional donde se hacen preguntas, transformando el juego en entretenimiento educativo.
Anoté todo esto en un papel, hice el programa de bloques y me puse a hacerlo. Primero hice el cuerpo principal, lo grabé, y después le fuí poniendo detalles para hacerlo más divertido, música, presentación, coronación, gráficos. Para mí lo principal es que todo sea prolijo y esté donde debe estar. Claro que el juego es un poco lento y nos tiene que gustar la temática, porque sino resulta aburrido. Lo pulí bien y cuando no tuvo ningún error lo grabé y se los mandé. Todo esto en no más de tres o cuatro días.
Me gusta mucho programar, y sobre todo cuando tengo un incentivo como este concurso. Y creo que es un poco lo que nos pasa a todos pero por suerte este tipo de cosas es la que nos despierta la mente y no deja caer a la MSX como una simple máquina de juegos, sobre todo a los que como a mí tienen la computadora como "hobby" y no con un uso técnico o profesional. Por eso siempre trato de contactarme con los usuarios de Paraná, y así compartimos información, trucos y secretos. Porque creo que la computadora nuclea a muchos y ayuda también a ser más sociable.
Me siento muy honrado de haber ganado el concurso de esta revista de nivel nacional. Y espero que sigan haciendo concursos para incentivar la imaginación de los usuarios.
Sin querer ser interesado o algo así les cuento que elijo la disketera como premio, muchas gracias.
La beca no la voy a poder aprovechar dado mi edad y ya que no podría de todas maneras estar en Buenos Aires por mucho tiempo, ya que sería bastante difícil.

Víctor Villarraza


Víctor Villarraza


Nicolás Moyano


Pedro Azcuí


Carlos Escobar

CHESSING

El programa desarrollado por Víctor Villaraza está dividido en dos secciones: entretenimiento y educativo.

Se trata de un entretenimiento que en algo se parece al tradicional "estanciero", pero que se desarrolla sobre un tablero de ajedrez. Los participantes pueden ser dos, y gana el que primero llega al casillero que se encuentra en el extremo opuesto al suyo en el tablero.

Para que el triunfo del ganador sea "total", deberá ser "coronado", cualidad que sólo podrá adquirirse si durante el transcurso de la competencia "compró" por lo menos siete "terrenos".

En cuánto a la parte gráfica, el juego consta de dos pantallas. La principal nos informa de nuestros adelantos y la secundaria se utiliza cuando se realizan preguntas.

La pantalla principal contiene el menú, el tablero, el tiro de los dados y la información de ambos jugadores.

Al comenzar el juego, se presenta el menú, con tres posibilidades: continuar el juego, instrucciones, y terminar el juego.

Respecto de los dados, el programa los arroja al azar, imprimiendo el número que sacó el jugador y avanza automáticamente la respectiva cantidad de casillas. Luego, pasa al turno del siguiente jugador.

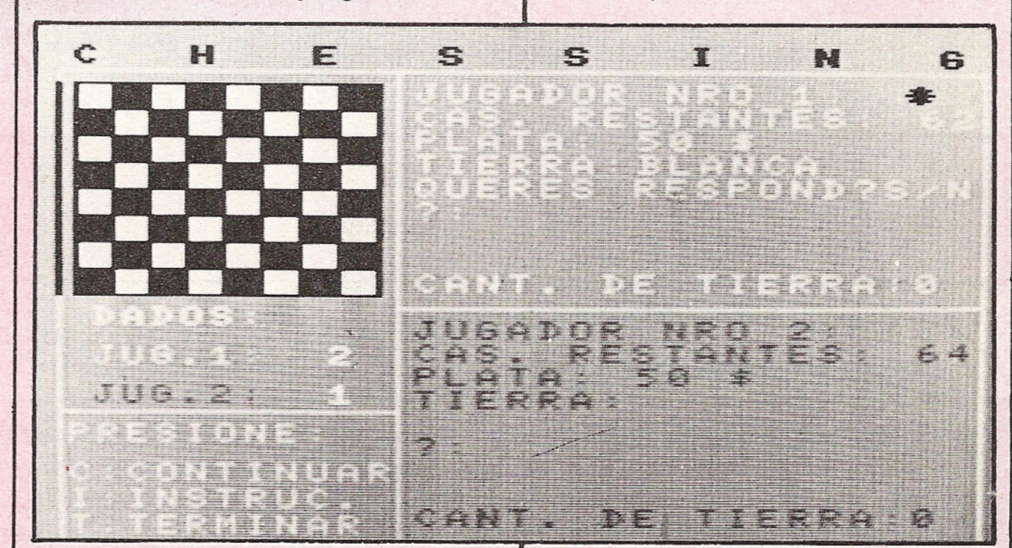
A cada uno de los participantes, por su parte, la pantalla le ofrece la siguiente información: cuántas casillas le faltan para la conclusión del juego (en principio, son 64), cuánto dinero dispone según los "negocios" que haya realizado, y el color del casillero en el cual está ubicado.

Si el cuadro es negro, estamos en territorio de la reina, que nos es favorable. Podemos comprarlo por cierta cantidad de dinero, vender los que hubiésemos comprado con anterioridad, o simplemente pasar sin hacer nada. Si, en cambio, el cuadro es blanco, estamos en territorio del rey, que nos dificulta el juego. Debemos contestar preguntas para poder seguir el juego o abonar una suma como peaje.

La situación se le presenta así al jugador número 1. En cambio, para el número 2, es totalmente a la inversa. Su territorio favorable es el blanco, mientras que en los casilleros negros debe contestar las preguntas o abonar el peaje.

El signo de interrogación, por su parte, indica la posibilidad de sacar una tarjeta que puede, según el caso, ser de buena o mala "suerte". Esta posibilidad está limitada a los casilleros que podemos comprar o vender, o si teníamos que responder una pregunta y no deseamos hacerlo.

Las "preguntas" tienen una pantalla especial, que aparece cuando seleccionamos esa posibilidad. Los temas sobre los que pueden tratar son "capitales" o "sinónimos". Las respuestas deben ser correctas, incluso debe cuidarse que no se cometan errores al tipear ya que no es posible corregir ni retroceder. Una ayuda importante que nos da el programa es la cantidad de letras que debe tener la palabra que se nos pide Por ejemplo, la pregunta "capital de Argentina", de 12 espacios,, deberá ser respondida "Buenos Aires", (el espacio entre palabras se cuenta como un caracter).



En caso que demos la respuesta correcta, ganamos dinero. Si, en cambio, nuestra contestación es incorrecta, perdemos una cantidad de efectivo mayor a la que podríamos haber ganado.

2° CONCURSO DE PROGRAMAS

auspiciado por TELEMATICA S.A. que proveerá los siguientes Premios:

PRIMER PREMIO

UN PERIFERICO

(a elección entre un monitor, una diskettera y una impresora).

UNA BECA

para trabajar en el Departamento de Investigación y Desarrollo de Telemática S.A.

SEGUNDO PREMIO

UN PERIFERICO

(a elección entre un monitor, una diskettera y una impresora).

ESPECIAL

Entre los programas recibidos, algunos de ellos podrán ser editados por Prosoft, reconociéndose los derechos de autor.

En caso de que el ganador no pueda utilizar la beca, será ofrecida a quien obtenga el segundo premio, y si éste tampoco pudiera aprovecharla se otorgará a alguno de los participantes del certamen que se hubiera destacado.

Se premiará el mejor software de cualquier clase (juegos, utilitarios, científico o comercial).

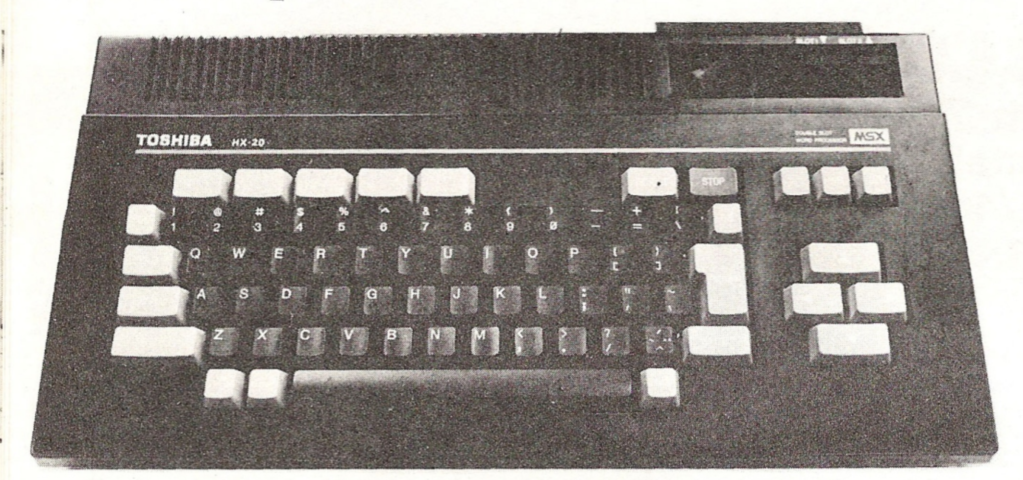
BASES: No sólo será indispensable que el programa enviado en caset ó disket funcione correctamente, sino que además debe cumplir con ciertas reglas:

- *Programación estructurada en bloques fácilmente diferenciables.*
- *Fácil seguimiento del mismo y detalle de éste como parte de su documentación. (Diagrama de bloques con los números de línea que los identifiquen).*
- *Aclaración y clara explicación de los algoritmos utilizados, deben figurar como parte de la documentación.*
- *Las variables y/o direcciones de memoria utilizados también se deben incluir en esta documentación.*
- *Listado de nemónicos assembler y la localización en memoria si es que se utiliza este tipo de lenguaje.*
- *Calidad y originalidad de gráficos, sonidos y pantallas de menú.*

Los trabajos deberán enviarse antes del 30 de julio próximo (cierre del certámen) a: Paraná 720, piso 5°, (1017) Capital Federal.

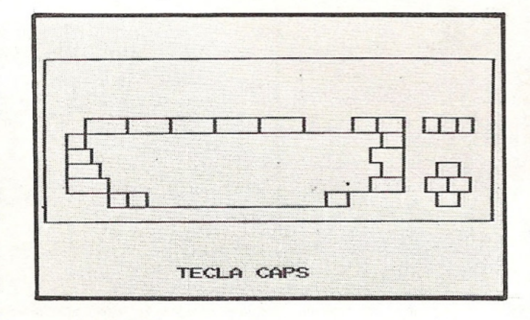
APLICACIONES PARA TOSHIBA

Para hacer más espontáneo el uso de la Toshiba HX-20 y marcando una relación más intensa entre el software y el hardware la máquina viene acompañada de un interesante y práctico paquete de programas de aplicaciones múltiples.



Con la adquisición de una microcomputadora hogareña comienza una nueva etapa en la planificación familiar, al buscar programas que cubran todas las expectativas.
La búsqueda no siempre es fácil: recorrer comercios, copiar todo lo que aparece sin saber ni siquiera lo que es.
Los resultados no siempre son los deseados y de a poco se pierden el entusiasmo por el uso de la máquina, o solamente reinan los juegos para los más chicos.
También hay que recurrir a manuales, algunos no muy claros o escritos en otros idiomas, para tratar de descifrar su funcionamiento.
La microcomputadora TOSHIBA HX-20 intenta solucionar estos inconvenientes al suministrar con la máquina seis cassettes con programas de los más variados tipos que cubren las necesidades esenciales.
No hace falta transformarse en un improvisado analista de sistemas y luego en un audaz programador con resultados no siempre ideales para poder realizar la Contabilidad de la casa y la Facturación del comercio.
También se podrá estudiar o repasar lecciones de inglés, generar la propia agenda o trabajos personales, sin conocer lenguajes ni computación. Y si el deseo es aprender a operar la máquina en una forma amena y fácil por medio de la instrucción programada, uno de los cassettes es la solución.
Y como este conjunto de programas tiene un fin netamente familiar y de aplicación inmediata, el último cassette es un juego por si queda algún ratito libre en el que no utilice la máquina el jefe de la familia.

CURSO DE BASIC



Este interesante programa es fundamental para aquél que vaya a dar el puntapié inicial (es una forma de decir) en el apasionante uso de la computadora.
Sin darnos cuenta podremos recorrer los laberintos de la máquina, conocer el perfecto uso de su teclas y comandos, y entendernos bastante bien en el lenguaje Basic.
El curso está integrado por 16 clases, a las cuales podremos repetir tantas veces como las necesitemos. Pero de vez en cuando se nos tomará un pequeño examen para ver si entendemos la explicación.

Lección	Tema
1	Teclado HX-20
2	Modo directo
3	Clases de datos
4	Modo programa
5	Comandos para escribir un programa
6	Comandos para ejecutar un programa
7	Escribir en pantalla
8	Entrada de datos
9	Toma de decisiones
10	Bucles
11	Subrutinas
12	Entradas de datos constantes
13	Grabación/lectura de programas
14	Modos de pantalla y color
15	Gráficos
16	Generar sonidos

Después de asistir a todas las clases nos vamos a dar cuenta de qué simple que es este mundo de la computación al cual ya estaremos incorporados.

CONTABILIDAD

```
1.-ENTRADA DE DATOS
2.-BUSQUEDA DE DATOS
3.-ANULACION DATOS
4.-BALANCE
5.-PREVISIONES
6.-DATOS BANCARIOS
7.-SALDOS BANCARIOS
8.-GRABACION FICHERO
9.-LECTURA FICHEROS
0.-DEFINICION CONCEPTOS
indique que opcion desea
```

Este programa ha sido pensado y desarrollado para que pueda ser utilizado en una función tan importante como poder llevar la contabilidad.
De una manera simple nos mantendremos al día en las entradas y salidas de un pequeño comercio o en el hogar y tendremos todos los movimientos del año acumulados en cassette.

También se podrán relacionar todos los movimientos de la contabilidad con hasta 9 cuentas bancarias, permitiendo en el momento deseado la obtención del resumen bancario y el saldo de disponibilidad a la fecha.
El programa posibilita manejar hasta 20 conceptos diferentes de entradas o salidas, e introducir hasta 250 movimientos.
En todo momento el programa indica los pasos a seguir a los que todavía encuentran la contabilidad como un misterio difícil de resolver.
De una manera muy simple y gracias a tan avanzada teconología electrónica, veremos en qué dirección se dirigen nuestro marchitos australes.

FACTURACION



La propuesta es muy interesante para pequeños comerciantes, a quienes este programa brindará óptimos resultados.
Está dividio en 2 menús principales. El primero nos permitirá dar altas, bajas, modificaciones, consultas, salidas, inventarios, etcétera.
El código del artículo estará comprendido entre 1 y 230 con una descripción que abarca 18 posiciones, una cantidad de stock actual, una cantidad mínima, su precio de venta y precio de compra.
También genera un inventario por pantalla o impresora a elección.
Con el segundo menú se podrá optar por listar pedidos, contabilidad, emitir pedidos, generar presupuestos personales o a clientes, etcétera.
Con el segundo menú se podrá optar por listar pedidos, contabilidad, emitir pedidos, generar presupuestos personales o a clientes, etc.
El listado de artículos en stock crítico marca la cantidad que se halla por debajo de la cantidad mínima especificada.
La contabilidad permite cuatro opciones que son: importe entrada (compras), que muestra la cantidad que gastamos en las compras. Muestra también el importe de salidas (ventas). La tercera opción es resultados.
También permite generar un balance total que es producto de listar los acumuladores.

BATALLA DE TANQUES

Es un Juego de habilidad guerrera, donde podremos mostrar la habilidad con las teclas y la rapidez visual.
Deberemos esquivar campos con minas en un típico escenario de hostilidades donde nuestro temple y sangre fría saldrán vencedores de esta tan ardorosa confrontación.
La tarea no será fácil, pero por lo menos no será riesgosa como lo es la guerra real.

BASE DE DATOS



Podremos con muy poco esfuerzo tener nuestra propia base de datos, guardando en ella direcciones, nombres, fechas, para el uso que más nos interese.
Incluso podemos participar en el diseño, dándole así características personales.
Con este programa la computadora será un aliado imprescindible para nuestra a veces frágil memoria.
Y así en un cassette tendremos una gran agenda sin preocuparnos demasiado por su extensión, y la actualización de la misma será rápida.

INGLES I



Es una interesante forma de aprender a repasar la lengua Sheaskespeareana, pero siempre con un tratamiento ameno.
Recorreremos una pequeña granja con las distintas facetas de su personaje central, un agricultor. Completaremos frases, dándonos tres oportunidades, y si ninguna es la buena el programa se encargará de completarlo en forma correcta.
El paseo es interesante, y poco cansador. Sin darnos cuenta estaremos aprendiendo.
Posee un marcado mensaje ecologista que nos demuestra que, con el tiempo, la quietud y armonía del campo se transforma y nos hallamos en ruidosas ciudades.
Al final nos calificará diciendo cómo fue nuestro desempeño en las clases y, si estuvimos flojitos, después de un recreo podemos reiniciar el estudio probablemente con más éxito.

UN ESCENOGRAFO EN LA COMPUTADORA

CLASE: ENTRETENIMIENTO

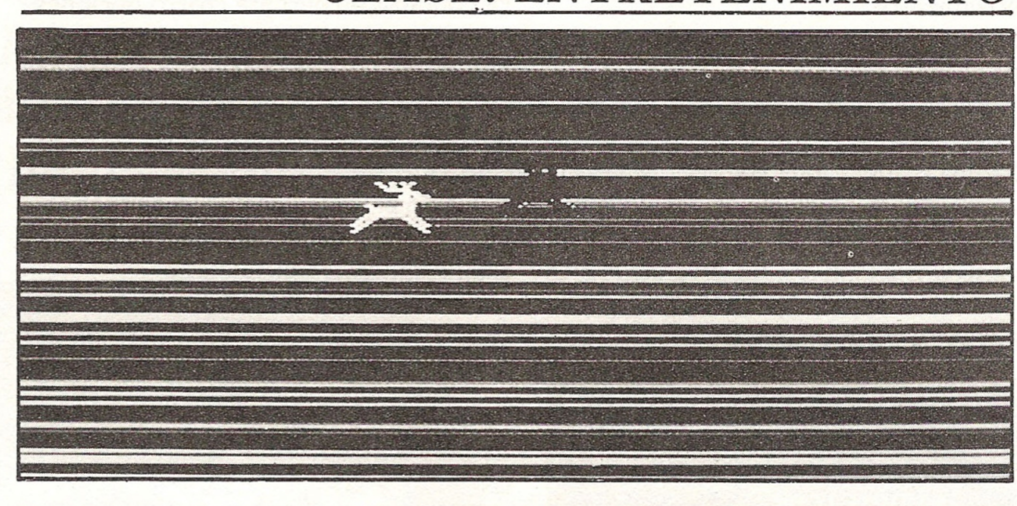
A través de este programa podremos observar una simulación de teatro en computadora.

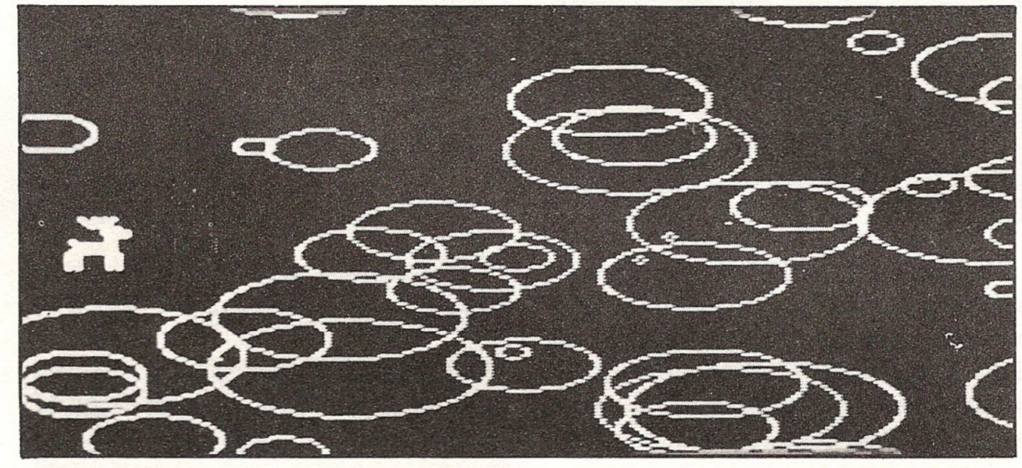
Pasan diferentes escenas, las cuales no siguen un argumento, pero son atractivas desde el punto de vista estético de las figuras.

Lo más destacable de este software son los efectos sonoros.

En este proyecto de obra de teatro, están inlcuidas varias piezas musicales. Todas siguen una línea de melodía sinfónica, como interludio.

Se definieron seis sprites, que formarán los personajes de esta obra.





VARIABLES IMPORTANTES:

A1$, A2$, A3$, A4$, A5$, A6$, A7$, A8$, A9$, B1$, B2$, B3$, B4$: contienen parte de la melodía.

XA, YA: coordenadas para los sprites.

ESTRUCTURA DEL PROGRAMA:

10-90: presentación
100-170: inicialización de sprites
180-290: inicialización de variables
300-4460: diferentes músicas
4470-4620: efecto 1
4630-4850: efecto 2
4860-4980: efecto 3
4990-5060: rutina de movimiento de los ciervos
5070-5090: datos para el sprite 0
5100-5120: datos para el sprite 1
5130-5150: datos para definir el sprite 2
5160-5180: datos para el sprite 3
5190-5210: datos del sprite 4
5220-5240: datos del sprite 5

```
10 COLOR 15,1,1:SCREEN 2,2
20 FOR P=2 TO 20 STEP 2
30 LINE (P,P)-((255-P),(191-P)),
P/2,B
40 NEXT P
50 OPEN"GRP:"AS#1
60 LINE(75,80)-(175,105),1,BF
70 PSET(85,90),1:PRINT#1,"TEATRO
 MSX"
80 FOR P=0 TO 2000:NEXT P
90 LINE(75,80)-(175,120),1,BF
100 RESTORE 5070
110 FOR I=0 TO 5
120 S$=""
130 FOR J=1 TO 32
140 READ R:S$=S$+CHR$(R)
150 NEXT J
160 SPRITE$(I)=S$
170 NEXT I
180 ' INICIALIZACION DE VARIABLE
S
190 DIM XA(5),YA(5)
200 XA(0)=50:XA(1)=100
210 YA(0)=50:YA(1)=100
220 XA(2)=60:XA(3)=30
230 XA(4)=30:XA(5)=10
240 A1$="o3ar16"
250 A6$="o3ar16ar16ar16a":A7$="o
5br16br16br16br16br16br16br16br16
":A8$="o5dr16dr16dr16dr16dr16dr16
dr16dr16":A9$="o4dr16dr16dr16d"
260 B1$="o5f+r16f+r16f+r16f+r16"
:B2$="o5dr16dr16dr16dr16":B3$="o5
fr16fr16fr16fr16":B4$="o5e-r16e-r
16e-r16e-r16"
270 LINE(0,0)-(255,191),1,BF
280 Z=0
290 ON INTERVAL=12 GOSUB 4470:IN
TERVAL ON
300 ' comienzo
310 PLAY"t50l16","t50l16","t50l1
6"
320 PLAY"v8o5f8.r16f8g64f32.f8ed
8","v8o4a8r8ar16a8.gf8","v8o3d4o4
d8o3d4."
330 PLAY"o5c+8.r16v12c+r16c+4r8"
,"r4v10o4gr16g4r8","o4e4v10o3er16
e4r8"
340 PLAY"v8g8.r16g8a64g32.g8fe8"
,"v8o4b-8r8b-16r16b-8.ag8","r2r4"
350 PLAY"o5d8.r16v12dr16d4r8","o
4f4v10dr16d4r8","r4v10o3f+r16f+4r
8"
360 PLAY"v10o6c4d32c32o5b32o6c32
dco5b-agf+","r2r4","r2r4"
370 PLAY"o5g4v12gr16g4r8","r4v10
dr16d4r8","r4v10o3b-r16b-4r8"
380 FOR I=1 TO 2:PLAY"v10o5g+4."
,"v10o4d64e64f32f8e16d8","v8o3b-4
.":NEXT I
390 PLAY"o5a4.r4r8","o5c+4.r4r8"
,A1$
400 PLAY"v10","v12","v10"
410 PLAY"o5b-ab-ab-8r8a8e8","o5e
4.r8e8c+8",A2$
420 PLAY"o5ag+ag+a8r8dr16dr16","
o5d4.r8o4ar16ar16",A2$
430 PLAY"o4b-ab-ab-8r8o5c+r16c+r
16","o5g4.r8er16er16",A2$
440 PLAY"o4ag+ag+a8r8ar16ar16","
o5f4.r8dr16dr16",A2$
450 PLAY"v12o5fr16fr16fr16f8v13g
fed","v12o5dr16dr16dr16d8r4","v10
o3b-r16b-r16b-r16b-4."
460 FOR I=1 TO 2:PLAY"v12o5fr16f
r16fr16f8v13gfed","v12r8o5dr16dr1
6d8r4","r8v10o3b-r16b-r16b-4.":NE
XT I
470 PLAY"v12","v10","v9"
480 PLAY"o5c+4.a4.","o4a4.o5e4."
,A1$
490 PLAY"v13o5b-4.o6c+4.","v11o5
f4.g4.",A1$
500 PLAY"v14o6d4.d4v13c8","v12o5
a4.a4v11g8",A1$
510 PLAY"v12o5b-4r8b-8a8g+8","v1
0o5f4r8fr16fr16fr16",A1$
520 PLAY"o5a4.r4r8","o5e4.r4r8",
A1$
530 PLAY"v11","v9","v6"
540 PLAY"o5b-4.g+4.","o5f4r8f4r8
",A1$
550 PLAY"v10o5a4.r4v9a8","v8o5e4
.r4v7e8",A1$
```

```
560 PLAY"o5g+4.b4.","o5f4.d4.",A
1$
570 PLAY"v8o5a4.r4r8","v6o5c+4.r
4r8",A1$
580 PLAY"o4g+64a64b64a4.a4.","r6
4r32o3a4.a4."
590 IF PLAY(0)=0 THEN 610
600 GOTO 590
610 INTERVAL OFF:COLOR ,,7
620 FOR P=0 TO 1:PUT SPRITE P,(0
,209):NEXT P
630 '
640 LINE(0,0)-(256,90),4,BF
650 LINE(0,90)-(256,192),3,BF
660 SN=0
670 '
680 ON INTERVAL=15 GOSUB 4630:IN
TERVAL ON
690 ' música
700 PLAY"t96","t96","t96"
710 I=0:GOSUB 3970
720 IF PLAY(0)=0 THEN 740
730 GOTO 720
740 INTERVAL OFF
750 FOR P=2 TO 3
760 PUT SPRITE P,(-20,-20)
770 NEXT P
780 LINE(0,0)-(256,192),4,BF
790 ON INTERVAL=16 GOSUB 4860:IN
TERVAL ON
800 '
810 PLAY"t901b","t901b","t901b":
J=0
820 J=J+1:PLAY"v13o6c+4.o5barbar
b","v10o5c+ec+ec+ec+","v8o3a1"
830 PLAY"o5br16r32o6c+32o5br16r3
2o6c+32o5br8r4","o5dg+dg+dg+dg+",
"o3a1"
840 PLAY"o6c+4r8c+dc+o5bo6c+","o
5egegegeg","o3a1"
850 PLAY"o6dr8dr8dr8r4","o5f+af+
af+af+a","o3a1"
860 PLAY"v11o5b4.o6c+16o5b16ag+f
+e","v8o5g+2r2","v8o5d2r2"
870 IF J=2 THEN 940
880 FOR K=1 TO 2
890 PLAY"o5ao6c+16o5b16","o5er8"
,"o5c+r8"
900 PLAY"o5b16a16g+16a16","r4","
r4"
910 NEXT K
920 PLAY"o5v12f+g+aa+bf+o6do5b",
"o4d2r2","r2r2"
930 PLAY"o5a4.bg+r4","o5c+ec+eo4
bo5ede","o4e2r2":GOTO 820
940 PLAY"o5a16r16a16r16o6c+edc+o
5ba+","o5e4r4r2","o5c+4r4r2"
950 PLAY"o5br8br8br8ar8","r4o4br
8br8br8","r4o4d+r8d+r8d+r8"
960 PLAY"l16","l8","l8":J=0
970 PLAY"v11o5g+ed+ef+ed+e","v11
o4b4r4","v11o4e4r4"
980 J=J+1:PLAY A5$,"o4bo5c+do4b"
,"O4G+ABG+"
990 PLAY"r16l32er32er32e","o5c+4
","o4a4"
1000 PLAY"er32er32er32el16","r8c
+","R4"
1010 PLAY A5$,"o5dc+o4ba","o4bag
+f"
1020 IF J=2 THEN 1040
1030 PLAY"r16o5ed+ef+ed+e","o4g+
4r4","o4e4r4":GOTO 980
1040 PLAY"o5e8r8g+8r8e8r8r4","o4
br8br8br8","o4g+r8g+r8g+r8"
1050 IF PLAY(0)=0 THEN 1070
1060 GOTO 1050
1070 INTERVAL OFF
1080 FOR P=4 TO 5
1090 PUT SPRITE P,(-20,-20)
1100 NEXT P
1110 LINE(0,0)-(256,192),4,BF
1120 LINE(0,90)-(256,192),3,BF
1130 ON INTERVAL=16 GOSUB 4630:I
NTERVAL ON
1140 '*************************
1150 PLAY"t96l8","t96l8","t96l16
v7":J=0
1160 PLAY"r2","r2",A6$
1170 PLAY"r4v10o6c+4","r4v10o5e4
",A6$
1180 FOR K=1 TO 2
1190 PLAY"o6c2","o5d+2",A6$
1200 PLAY"o6c+r8c+r8","o5er8er8"
,A6$
1210 NEXT K
1220 PLAY"v10o6e2","v10o5g2",A6$
1230 PLAY"o6dr8dr8","o5f+r8f+r8"
,A6$
1240 PLAY"v11o5bo6f+ed","v11o5g+
r8r4","o3a2v8"
1250 PLAY"o6c+4v11o5c+4","o5a4v1
1o4e4",A6$
1260 FOR K=1 TO 2
1270 PLAY"o5c2","o4d+2",A6$
1280 PLAY"o5c+r8c+r8","o4er8er8"
,A6$
1290 NEXT K
1300 PLAY"v11o5e2","v11o4g2",A6$
1310 PLAY"o5dr8dr8","o4f+r8f+r8"
,A6$
1320 PLAY"v12o4bo5f+ed","v12o4g+
r8r4","o3a2"
1330 PLAY"v13o5c+l16er16er16er16
","o4ar8r4","o3a4r4"
1340 J=J+1:K=0
1350 K=K+1:PLAY"r16o5ed+ef+ed+e"
,"v10o5dr8r4","v10o4br8r4"
1360 PLAY"o5g+ebg+e8r8","r4r8o5d
","r4r8o4b"
1370 PLAY"r16o5ed+ef+ed+e","o5c+
4r4","o4a4r4"
1380 IF K=2 THEN 1400
1390 PLAY"o5aed6c+o5ae8r8","r4r8
o5c+","r4r8o4a+":GOTO 1350
1400 PLAY"o5aeo6c+o5aebr8","r2",
"r2"
1410 PLAY"r4l8v10o5a+r8a+r8a+r8"
,"v8o4f+o5ec+ec+ec+e","r2r2v8"
1420 PLAY"o5bo6c+dc+ed+do5b","o5
f+4r4r2","o5d4r4r2"
1430 PLAY"o6c+8.o5b16a2g+4","o5c
+ec+ec+ede","o3a2e4r4"
1440 IF J=2 THEN 1470
1450 PLAY"v12l16o5ac+def+ef+g+",
"O5C+16R16R8R4","O3A4R4"
1460 PLAY"o5abo6c+o5bag+f+e","r2
","r2":GOTO 1340
1470 PLAY"l4","l4","l16v9"
1480 FOR K=1 TO 2
1490 PLAY"v12o5ar4","v12o5c+r4",
A6$
1500 PLAY"r4o5v11a","r4o5v11f+",
A6$
1510 PLAY"o5g+f+","o5ed",A6$
1520 PLAY"o5ed+","o5c+c",A6$
1530 PLAY"o5e2","o5c+2",A6$
1540 PLAY"o5c+r8l8g","o4ar8l8o5c
+",A6$
1550 PLAY"v12o5f+f+16e16f+a","v1
```

```
2o5dr8dr8",A6$
1560 PLAY"v13o5g+ebg+l4","v13o4b
r8br8l4",A6$
1570 NEXT K
1580 FOR K=1 TO 2
1590 PLAY"l8v14o5a4v15o6c+4.v12o
5bag+","l8v14o5er8v15e4.v12dc+e",
"v12o3a8r8a4r8r4"
1600 PLAY"o5f+edc+v15f+4g+4","o5
dc+o4bav15o5d4o4b4","r2o3d4e4"
1610 NEXT K
1620 PLAY"l16v8","l16v8","v9"
1630 FOR K=1 TO 2
1640 FOR M=1 TO 2:PLAY"r8o5ar16a
4","r8o5c+r16c+4","o3a4.o4e8":NEX
T M
1650 FOR M=1 TO 2:PLAY"r8o5br16b
4","r8o5dr16d4","o3a4.o4g+8":NEXT
 M
1660 NEXT K
1670 FOR K=1 TO 2
1680 PLAY"r8o5ar16a4","r8o5c+r16
c+4","o3a4.o4e8"
1690 PLAY"r8o5br16b4","r8o5dr16d
4","o3a4.o4g+8"
1700 NEXT K
1710 PLAY"l8r4v13o5ar8ar8o6c+r8"
,"l8r4v13o5er8er8er8","l8o3ar8v13
ar8ar8ar8"
1720 PLAY"o5a2r2","o5e2r2","o3a2
r2"
1730 IF PLAY(0)=0 THEN 1750
1740 GOTO 1730
1750 INTERVAL STOP
1760 LINE(0,0)-(256,90),4,BF
1770 LINE(0,100)-(256,192),12,BF
1780 INTERVAL ON
1790 ' ************************
1800 PLAY"l4","l4","l4"
1810 FOR K=1 TO 4:PLAY"v10o5a","
v10o5e-","v10o4f":NEXT K
1820 PLAY"o5a2r2","o5e-2r2","o4f
2r2"
1830 FOR K=1 TO 4:PLAY"o5b-","o5
f","o4d-":NEXT K
1840 PLAY"o5b-2r2","o5f2r","o4d-
2r2"
1850 FOR K=1 TO 4:PLAY"o6c","o5g
-","o4e-":NEXT K
1860 PLAY"o6c2r2","o5g-2r","o4e-
2r2"
1870 PLAY"l16","l8","l8"
1880 FOR K=1 TO 2
1890 FOR M=1 TO 2
1900 PLAY"v15o6d-2.d-8","V13R8O5
FG-FG-FG-F","V13R8O4A-B-A-B-A-B-A
-"
1910 NEXT M
1920 PLAY"v13o5e-g-o6co5g-e-g-o6
co5g-","r4o4a-4","r2"
1930 PLAY"o5e-g-o6co5g-e-g-o6co5
g-","o4a4a-4","r2"
1940 PLAY"o5e-g-o6co5g-e-g-o6co5
g-e-g-o6co5g-e-g-o6co5g-","o4a4a-
4a4a-","r1"
1950 NEXT K
1960 FOR K=1 TO 16
1970 PLAY"v15o6d-r16","v13o5f16r
16","v13o5d-16r16"
1980 NEXT K
1990 PLAY"l16","l16","l8"
2000 PLAY A7$,A8$,"o4g+2g+4.g+16
"
2010 PLAY A7$,A8$,"o4g+2g+4.g+16
"
2020 PLAY A7$,A8$,"r4o4e4g+4e4"
2030 FOR K=1 TO 2
2040 FOR K=1 TO 2
2050 PLAY"o4ao5c+ea","r4","r4"
2060 FOR M=1 TO 3
2070 PLAY"o4ao5c+ea","l8o3ar8","
r4"
2080 NEXT M
2090 NEXT K
2100 FOR K=1 TO 2
2110 PLAY"o4go5c+eb-","r4","r4"
2120 FOR M=1 TO 3
2130 PLAY"o4go5c+eb-","o3ar8","r
4"
2140 NEXT M
2150 NEXT K
2160 FOR K=1 TO 2
2170 PLAY"o4fao5da","r4","r4"
2180 FOR M=1 TO 3
2190 PLAY"o4fao5da","o3ar8","r4"
2200 NEXT M
2210 NEXT K
2220 FOR K=1 TO 2
2230 PLAY"o4eb-o5c+g","r4","r4"
2240 FOR M=1 TO 3
2250 PLAY"o4eb-o5c+g","o3ar8","r
4"
2260 NEXT M
2270 NEXT K
2280 PLAY"l8","l8","l8"
2290 FOR K=1 TO 2
2300 PLAY"r8f16r16f4r8f16r16f4",
"r8o5d16r16d4r8d16r16d4","o4d4.e8
f4.g"
2310 PLAY"r8o5fr8fr8fr8fv8","r8o
5dr8dr8dr8dv8","o4a4b-4g+4a4v10"
2320 NEXT K
2330 PLAY"r8f16r16f4r8v10f16r16f
4","r8o5d16r16d4r8v10d16r16d4","o
4b-4.v14f8d4.f"
2340 PLAY"r8o5fr8fr8fr8f","r8o5d
r8dr8dr8d","o3b-4o4d4f4b-4"
2350 PLAY"v8r8f16r16f4r8v10f16r1
6f4","v8r8o5d16r16d4r8v10d16r16d4
","o4a4.v10f8d4.f"
2360 PLAY"r8o5fr8fr8fr8fv6","r8o
5dr8dr8dr8dv6","o3a4o4d4f4a4"
2370 PLAY"v10r8f16r16f4r8v10f16r
16f4","v10r8o5d16r16d4r8v10d16r16
d4","v14o4g+4.v12ab-4.ab"
2380 PLAY"v11r8o5fr8fr8fr8f","v1
1r8o5dr8dr8dr8d","v14o4g+4a4b-4g+
"
2390 PLAY"l8v13o5e4r4","l8v12o5c
4r4","l16v10o4ar16o3ar16ar16a":J=
0
2400 J=J+1:PLAY"r4o5ar8","r4o5er
8",A6$
2410 PLAY"o5br8br8","o5g+r8g+r8"
,A6$
2420 PLAY"o6c+r8c+r8","o5gr8gr8"
,A6$
2430 PLAY"o6dr8dr8","o5ar8ar8",A
6$
2440 PLAY"o6d4r4","o5a4r4",A6$
2450 PLAY"o5b2","o5f2",A6$
2460 PLAY"o5g+2","o5f2",A6$
2470 IF J=2 THEN 2490
2480 PLAY"o5a2","o5e2",A6$:GOTO
2400
2490 PLAY"l16o5ar16o4ag+ar16ag+"
,"l16o5c+8r8r4",A6$
2500 PLAY"o4a4r4","r2",A6$
2510 PLAY"v11","v10","v8"
2520 PLAY"o4gr16gf+gr16gf+","r2"
,A6$
2530 PLAY"o4g4r4","r2",A6$
2540 PLAY"v10","v8","v6"
2550 PLAY"o4f+r16f+ff+r16f+f+","
o4ar16ar16ar16ar16",A6$
2560 PLAY"o4f+4r4","o4a4r4",A6$
2570 PLAY"v9","v7","v6"
2580 PLAY"o4er16ed+er16ed+","o4g
r16gr16gr16gr16",A6$
2590 PLAY"o4e4r4","o4g4r4",A6$
2600 GOSUB 3970
2610 ' ************************
2620 PLAY"l8v11","l8v11","l16v8"
2630 PLAY"r2","r2",A9$
2640 PLAY"r4o5a4","r4o5f+4",A9$
2650 FOR K=1 TO 2
2660 PLAY"o5g+2","o5f2",A9$
2670 PLAY"o5ar8ar8","o5f+r8f+r8"
,A9$
2680 NEXT K
2690 PLAY"o6c2","o5a2",A9$
2700 PLAY"o5br8bo6d","o5gr8gr8",
A9$
2710 PLAY"o6c+edc+","r2","o3a2v1
0"
2720 PLAY"o6d4v12o4a4","o5f+4v12
o4f+4",A3$
2730 FOR K=1 TO 2
2740 PLAY"o4g+2","o4f2",A3$
2750 PLAY"O4AR8AR8","O4F+R8F+R8"
,A3$
2760 NEXT K
2770 PLAY"o5c2","o4a2",A3$
2780 PLAY"o4br8bo5d","o4gr8gr8",
A3$
2790 PLAY"v12o5c+age","v12o4g2",
"v12o3a2"
2800 PLAY"o5d4r4","o4f+4r4","o3d
4r4":J=0
2810 J=J+1:PLAY"l16","l16","l16"
2820 FOR K=1 TO 2
2830 PLAY"v13o5eag+abag+a","v10o
5c+8r8r4","r2"
2840 PLAY"o6c+o5ag+agaea","r4o5c
+r16c+r16","r2"
2850 PLAY"o5f+ag+abag+a","o5d8r8
r4","r2"
2860 PLAY"o6do5ag+af+af+a","r4o5
dr16dr16","r2"
2870 NEXT K
2880 IF J=2 THEN 2900
2890 FOR M=1 TO 16:PLAY"v13o5f+"
,"v13o5d+","v13o3b":NEXT M:GOTO 2
910
2900 FOR K=1 TO 4:PLAY"o3bo4ao5d
+f+","r4","r4":NEXT K
2910 PLAY"v13o5ed+f+egf+ag","v10
o4b2","V10O4G2"
2920 PLAY"o5e8o6c+o5ba+bge","o4b
2","o4g2"
2930 PLAY"v12o5f+8.ed2c+4","v12o
4a8.gr2o4g4","v10l8r8o3ao4f+o3ao4
eo3ar8a"
2940 IF J=2 THEN 2960
2950 PLAY"o5d4a8bagr16gr16f+r16f
+r16","o4f+4o5f+r16f+r16ar16er16d
r16dr16","o3d4r4r2":GOTO 2810
2960 PLAY"l4","l4","l16v10":J=0
2970 PLAY"o5dr4","o4f+r4",A3$
2980 J=J+1:PLAY"r4v11o5d","r4v11
o4b",A3$
2990 PLAY"o5c+o4b","o4ag",A3$
3000 PLAY"o4ag+","o4f+f",A3$
3010 PLAY"o4ar4","o4f+r4",A3$
3020 PLAY"v13o5c2","v13o4a2",A3$
3030 PLAY"v12o4br4","v12o4gr4",A
3$
3040 PLAY"v13o5g2","v13o5c+2",A3
$
3050 IF J=2 THEN 3070
3060 PLAY"v12o5f+r4","v12o5dr4",
A3$:GOTO 2980
3070 PLAY"l8","l8","L4"
3080 PLAY"v14o5f+4a4.g+aa+","v13
o5d4f+4.ff+16r16f+","v11o3do4do3d
r4"
3090 PLAY"o5ba+ba+b4o6c+4","o5gf
+gf+gr8gr8","o4g2r4o3a"
3100 PLAY"o6d4o5a4.g+aa+","o5f+r
8f+4.ff+16r16f+","o3do4do3dr4"
3110 PLAY"o5ba+ba+b4o6c+4.","o5g
f+gf+gr8g4.","o4g2r4o3a."
3120 IF PLAY(0)=0 THEN 3140
3130 GOTO 3120
3140 INTERVAL OFF
3150 LINE(0,0)-(256,192),4,BF
3160 SN=SN+1
3170 ON INTERVAL=17 GOSUB 4630:I
NTERVAL ON
3180 PLAY"r1","r1","r1"
3190 ' ************************
3200 FOR K=1 TO 2
3210 FOR M=1 TO 2
3220 PLAY"v10r8r16o5d32r32d4","v
10r8r16o4f+32r32f+4","v8o3d4.a8"
3230 NEXT M
3240 FOR M=1 TO 2
3250 PLAY"r8r16o5e32r32e4","r8r1
6o4g32r32g4","o3d4.o4c+8"
3260 NEXT M
3270 NEXT K
3280 PLAY"l16","l16","l8":J=0
```

```
3290 PLAY"v13r4o5f+r16f+r16","v1
3r4o5dr16dr16","v15o3d4o4dc+"
3300 J=J+1
3310 FOR K=1 TO 2
3320 PLAY B1$,B2$,"o3ba+ba+"
3330 PLAY B1$,B2$,"O3BF+O4DC+"
3340 NEXT K
3350 PLAY B1$,B2$,"o3bagf+"
3360 FOR K=1 TO 4:PLAY "o5er16",
"o4br16","o3g16r16":NEXT K
3370 IF J=2 THEN 3400
3380 FOR K=1 TO 4:PLAY"o5er16","
o5c+r16","o3a16r16":NEXT K
3390 PLAY B1$,B2$,"o3d4o4dc+":GO
TO 3300
3400 FOR K=1 TO 4:PLAY"o5gr16","
o5c+r16","o3a16r16":NEXT K
3410 PLAY"v11r8o5fr16fr16fr16","
v11r8o5dr16dr16dr16","v12o3b-4r4"
3420 PLAY B3$,B2$,"r4o3b-4":J=0
3430 J=J+1:PLAYB3$,B4$,"o3a4o4c4
"
3440 PLAY B3$,B4$,"o3a4f4"
3450 IF J=2 THEN 3480
3460 PLAY B3$,B2$,"o3b-4r4"
3470 PLAY B3$,B2$,"r8o4dco3b-":G
OTO 3430
3480 PLAY B3$,B2$,"v12o3b-2"
3490 PLAY B3$,B2$,"r4o3a4"
3500 PLAY"v12o5br16br16br16br16"
,"v12o5fr16fr16fr16fr16","v13o3g+
r8g+r8"
3510 PLAY"v13o5br16br16v14br16br
16","v13o5fr16fr16v14fr16fr16","v
14o3g+r8g+r8"
3520 FOR K=1 TO 4:PLAY"v15o4ao5d
f+a","v15o3a4","r4":NEXT K
3530 FOR K=1 TO 4:PLAY"o4ao5ego6
c+","o3a4","r4":NEXT K
3540 PLAY"v13r8o6dr16dr16dr16","
v13r8o5f+r16f+r16f+r16","14v13o4d
r4"
3550 PLAY"r8o6d8r8d8","r8o5f+8r8
f+8","o5dv12o4a+"
3560 PLAY"r8o6dr16dr16dr16","r8o
5f+r16f+r16f+r16","o4br4"
3570 PLAY"r8o6d8r8d8","r8o5f+8r8
f+8","o4ba"
3580 PLAY"r8o6dr16dr16dr16","r8o
5er16er16er16","o4gr4"
3590 PLAY"r8o6d8r8d8","r8o5e8r8f
8","o4gg+"
3600 PLAY"r8o6dr16dr16dr16","r8o
5f+r16f+r16f+r16","o4a2"
3610 PLAY"r8o6c+r16c+r16c+r16","
r8o5gr16gr16gr16","o3a2"
3620 IF PLAY(0)=0 THEN 3640
3630 GOTO 3620
3640 INTERVAL OFF:COLOR ,,1
3650 FOR P=2 TO 3
3660 PUT SPRITE P,(0,209)
3670 NEXT P
3680 LINE(0,0)-(256,192),1,BF
3690 FOR P=1 TO 10
3700 LINE(0,15*P)-STEP(256,2*(P-
1)),9,BF
3710 NEXT P
3720 PLAY"r1","r1","r1"
3730 XA(4)=30:XA(5)=10
3740 ON INTERVAL=15 GOSUB 4990:I
NTERVAL ON
3750 FOR P=1 TO 10
3760 PLAY"t9014v12","t90v1214","
t90v1214"
3770 FOR K=1 TO 2
3780 FOR M=1 TO 2
3790 PLAY"r8r16o5d32r32d4","r8r1
6o4f+32r32f+4","o3d4.a8"
3800 NEXT M
3810 FOR M=1 TO 2
3820 PLAY"r8r16o5e32r32e4","r8r1
6o4g32r32g4","o3d4.o4c+8"
3830 NEXT M
3840 NEXT K
3850 PLAY"v13","v13","v13"
3860 FOR K=1 TO 2
3870 PLAY"r8r16o5d32r32d4","r8r1
6o4f+32r32f+4","o3d4.a8"
3880 PLAY"r8r16o5e32r32e4v14","r
8r16o4g32r32g4v14","o3d4.o4c+8v14
"
3890 NEXT K
3900 PLAY"r4o5d8r8d8r8d8r8","r4o
4f+8r8f+8r8f+8r8","v15o3do4do3af+
"
3910 PLAY"v15o5dr4o6dr4","v14o4f
+r4o5f+r8","o3dr4dr4"
3920 PLAY"t60s0m48000o6d1r2","t6
0s0m48000o5f+1r2","t60s0m48000o3d
1r2"
3930 IF PLAY(0)=0 THEN 3950
3940 GOTO 3930
3950 END
3960 ' ************************
3970 '*************************
3980 I=I+1:J=0
3990 J=J+1
4000 FOR K=1 TO 4:PLAY"v15o4dr16
dc+","r4","v13o3d4":NEXT K
4010 PLAY"v1514","v1214","v10"
4020 PLAY"o4d2","r2",A3$
4030 PLAY"r4v12o5d","r4o4b",A3$
4040 PLAY"o5c+o4a","o4af+",A3$
4050 PLAY"o4gb","o4eg", A3$
4060 PLAY"o4a2","o4f+2",A3$
4070 PLAY"o4f+r4","o4dr4","o3d8e
8f+8g8"
4080 PLAY"o5a8g+8g2f+","v11o5c+8
d8e2d","v12o3a4.b8o4c+8o3a8o4d8o3
d8"
4090 PLAY"o5e8f+8g2f+","o5c+8d8e
2d","o3a4.b8o4c+3o3a8o4d8f+8"
4100 IF J=2 THEN 4140
4110 PLAY"l10o5ed+f+egf+age8o6c+
o5bagf+e","r2r4l16","v10o4g1"
4120 PLAY"v15o5d4e32d32c+32d32e8
","v11o4f+4.g8",A3$
4130 PLAY"o5c+4r4","o4e4r4","o3a
r16ar16g8e8":GOTO 3990
4140 PLAY"l16v11","l16v9","v12"
4150 FOR K=1 TO 2
4160 PLAY"o5e8f+32e32d+e8g8c+8d3
2c+32o4bo5c+8e8","r2r2","v10o4g2a
2"
4170 PLAY"o5d8e8f+8g8g+8a8f+8d8"
,"v10o4f+8r8r2r4","o3d2r2"
4180 NEXT K
4190 IF I=2 THEN 4220
4200 PLAY"o5e8f+32e32d+e8g8c+8d3
2c+32o4bo5c+8e8","r2r2","v10o4g2a
2"
4210 PLAY"v15o5d8r8d8r8d8er8r4",
"v12o4f+8r8f+8r8f+8r8","v12o3d8r8
d8r8d8r8r4":GOTO 4240
4220 PLAY"o5er16er16g8b8c+r16c+r
16a8c+8","r2r2","v10o4g2a2"
4230 PLAY"o5d8r8d8r8d8r8r4","o4f
+8r8f+8r8f+8r8","o3d8o4d8o3a8f+8d
8f+8a8o4d8"
4240 PLAY"l16v11","l16v9","v12"
4250 FOR K=1 TO 2
4260 PLAY"v13r16o5df+o6dr16o5df+
o6d","r2","v12o3f+4o4d4"
4270 PLAY"r16o5df+o6dr16o5df+o6d
","r2","o4f+4d4"
4280 FOR M=1 TO 4:PLAY"o6c+8.r16
","v9o5g8.o4a","o3a4":NEXT M
4290 NEXT K
4300 PLAY"v15l24o6dr24dr24dr24dr
24dr24dr24","v10o5f+2","v12o4d2"
4310 PLAY"o6dr24dr24dr24c+r24c+r
24c+r24","r4g4","r4a+4"
4320 PLAY"o5br24br24br24br24br24
br24","o5f+2","o4b2"
4330 PLAY"o5br24br24br24ar24ar24
ar24","r4d4","r4f+4"
4340 PLAY"o5gr24gr24gr24gr24gr24
gr24","o5d2","o4g2"
4350 PLAY"o5ar24ar24ar24ar24ar24
ar24","o5f6r12f6r12","o4a6r12a6r1
2"
4360 PLAY"o5g+r24g+r24g+r24g+r24
g+r24g+r24","o5f6r12f6r12","o4b-6
r12b-6r12"
4370 PLAY"v13","v10","v12"
4380 FOR K=1 TO 2
4390 PLAY"r8o5ar16a8r16a","r8o5e
r16e8r16e",A4$
4400 PLAY"r8o6c+r16c+8r16c+","r8
o5gr16g8r16g",A4$
4410 PLAY"r8o6dr16d8r16d","r8o5f
+r16f+8r16f+",A4$
4420 PLAY"r8o5br16b8r16b","r8o5f
r16f8r16f","o3a4.o4g+8"
4430 NEXT K
4440 IF I=2 THEN 4460
4450 PLAY"v15o5a4o6c+4o5a4r4","l
8v13o5er8er8er8r4","l8v13o3ar8ar8
ar8r4":RETURN
4460 PLAY"l8v15r4o5ar8ar8r4","l8
v13r4o5er8er8r4","l8v13o3ar8ar8ar
8r4":RETURN
4470 ' ************************
4480 X1=INT(RND(1)*256)
4490 Y1=INT(RND(1)*192)
4500 PSET(X1,Y1),15
4510 ' ************************
4520 Z=1-Z
4530 FOR P=0 TO 1
4540 XA(P)=XA(P)+(P*16)
4550 YA(P)=YA(P)+(P*16)
4560 PUT SPRITE P,(XA(P),YA(P)),
15,Z
4570 XA(P)=XA(P)+8
4580 YA(P)=YA(P)+5
4590 IF XA(P)>255 THEN XA(P)=-20
4600 IF YA(P)>191 THEN YA(P)=-20
4610 NEXT P
4620 RETURN
4630 ' ************************
4640 IF SN=1 THEN 4740
4650 ' ************************
4660 X2=INT(RND(1)*124)
4670 Y2=INT(RND(1)*88)
4680 C2=INT(RND(1)*12)+2
4690 FOR P=0 TO 1
4700 LINE(X2+P*124,Y2+100)-STEP(
6,2),C2,BF
4710 LINE(X2+P*124,Y2)-STEP(8,0)
,7
4720 NEXT P
4730 GOTO 4780
4740 ' ************************
4750 Y4=INT(RND(1)*192)
4760 C4=INT(RND(1)*12)+2
4770 LINE(0,Y4)-STEP(256,0),C4
4780 ' ************************
4790 Z=1-Z
4800 FOR P=2 TO 3
4810 PUT SPRITE P,(XA(P),65+5*P)
,(1+14*(P-2)),2+Z
4820 XA(P)=XA(P)+10
4830 IF XA(P)>255 THEN XA(P)=-20
4840 NEXT P
4850 RETURN
4860 ' ************************
4870 X3=INT(RND(1)*256)
4880 Y3=INT(RND(1)*192)
4890 R=INT(RND(1)*30)
4900 CIRCLE(X3,Y3),R,7,,,.5
4910 ' ************************
4920 Z=1-Z
4930 FOR P=4 TO 5
4940 PUT SPRITE P,(XA(P),65+5*P)
,(1+14*(P-4)),4+Z
4950 XA(P)=XA(P)+5
4960 IF XA(P)>255 THEN XA(P)=-20
4970 NEXT P
4980 RETURN
4990 ' ************************
5000 Z=1-Z
5010 FOR P=4 TO 5
5020 PUT SPRITE P,(XA(P),151),1,
4+Z
5030 XA(P)=XA(P)+5
5040 IF XA(P)>255 THEN XA(P)=-20
5050 NEXT P
5060 RETURN
5070 ' ************************
5080 DATA 0,0,8,24,56,4,3,3,1,0,
3,15,31,62,126,252
5090 DATA 1,3,7,15,31,31,62,184,
240,192,160,144,12,12,2,1
5100 ' ************************
5110 DATA 2,3,11,25,56,4,227,115
,63,31,7,0,0,0,0,0
5120 DATA 0,0,128,192,192,224,22
4,224,224,192,160,16,12,12,2,1
5130 ' ************************
5140 DATA 4,2,1,0,0,0,0,16,15,15
,31,40,80,160,64,128
5150 DATA 168,144,240,28,55,31,2
4,24,248,248,232,20,10,5,2,1
5160 ' ************************
5170 DATA 4,2,1,0,0,0,0,16,15,15
,31,40,20,10,5,2
5180 DATA 168,144,240,28,55,31,2
4,24,248,248,232,20,40,80,160,64
5190 ' ************************
5200 DATA 4,2,1,0,0,0,0,16,15,15
,31,40,40,40,40,40
5210 DATA 168,144,240,28,55,31,2
4,24,248,248,232,20,20,20,20,20
5220 ' ************************
5230 DATA 4,2,1,0,0,0,0,16,15,15
,31,40,40,68,68,68
5240 DATA 168,144,240,28,55,31,2
4,24,248,248,232,20,20,34,34,34
```

METODOS PARA ORDENAR DATOS

Ordenar datos en forma ascendente o descendente es una operación frecuente en computación. Veremos aquí dos de los métodos básicos.

Al trabajar con manipulación de datos, frecuentemente nos vemos obligados a ordenarlos en forma ascendente, descendente o bien alfabéticamente.
Para manejar datos de esta forma, ya sean números o palabras, es conveniente ir guardandolos a medida que se ingresan, en un vector o matriz lineal.
A este tipo de almacenamiento lo podemos pensar como si guardásemos datos en cajas ubicadas una a continuación de la otra, como vemos en la figura 1.
Para facilitar el entendimiento de esta nota, supongamos que observamos esta hilera de cajas desde arriba, pudiendo ver el contenido de cada casillero (ver figura 2).
Cada caja tiene un número que la identifica. Este número coincide con el orden correspondiente a dicha caja. Es decir, la primera caja llevará el número "1", la segunda el número "2", la tercera el "3" y así sucesivamente hasta la última.
Supongamos que tenemos que trabajar vector como el de la figura 2.
Como podemos observar, este vector tiene sólo ocho elementos, cifras en este caso.
Pero la ubicación de los datos no sigue ninguno de los órdenes mencionados anteriormente.
Si necesitamos tener estos datos ordenados en forma ascendente, por ejemplo, deberíamos ir pasando los contenidos de las cajas, hasta conseguir obtener los datos de la forma deseada.
Para este procedimento existen varias formas de realización.
Básicamente son los métodos que responden a los nombres:
*- selección I
*- selección II
*- selección a derecha
*- burbujeo
*- shell

FIGURA 1

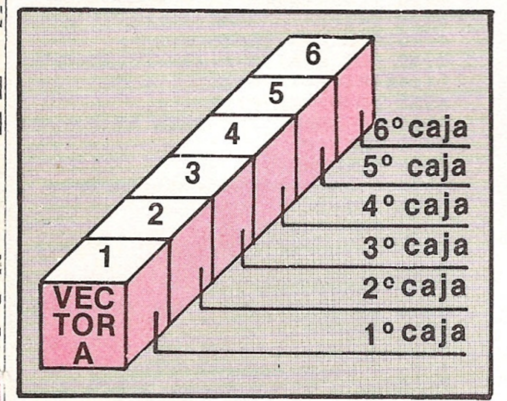


FIGURA 2

13	casillero 1
10	casillero 2
2	casillero 3
15	casillero 4
8	casillero 5
0	casillero 6
9	casillero 7
1	casillero 8

FIGURA 3

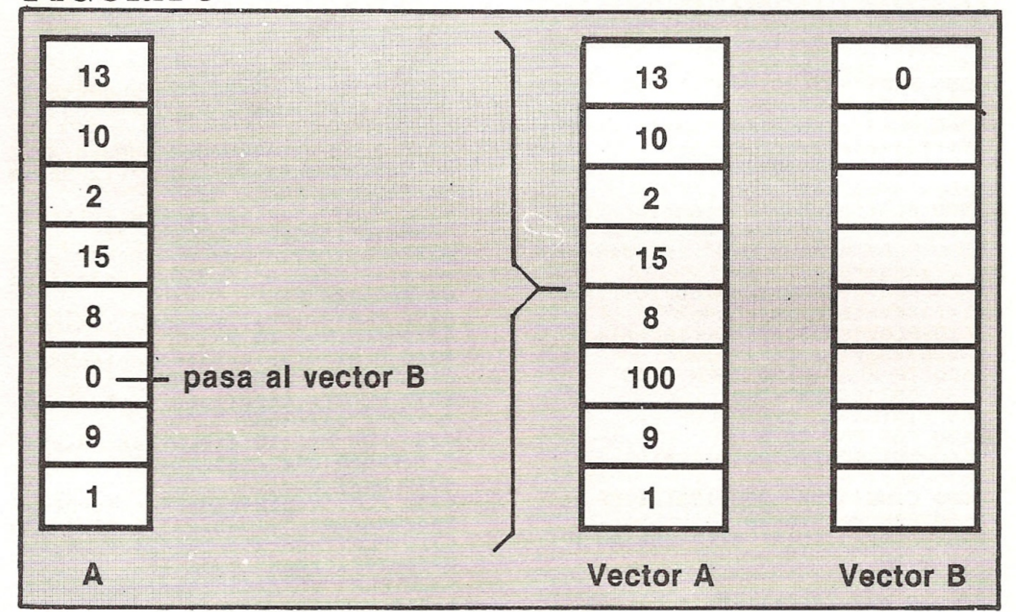
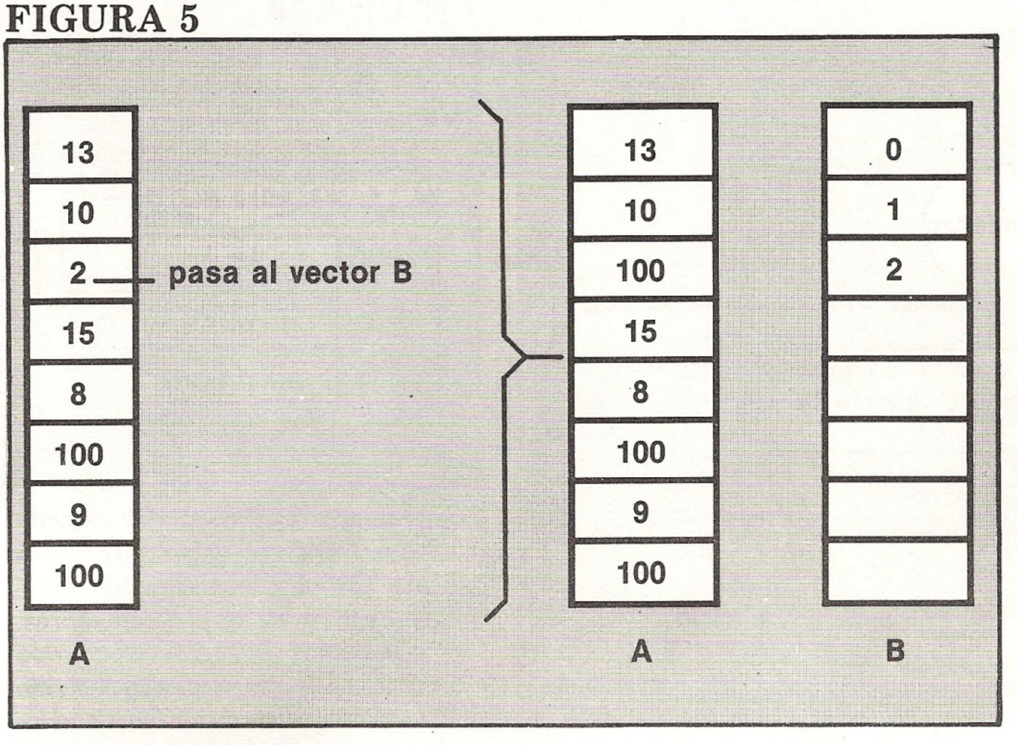


FIGURA 4

A		A	B
13		13	0
10		10	1
2		2	
15		15	
8		8	
100		100	
9		9	
1	pasa al vector B	100	

FIGURA 5





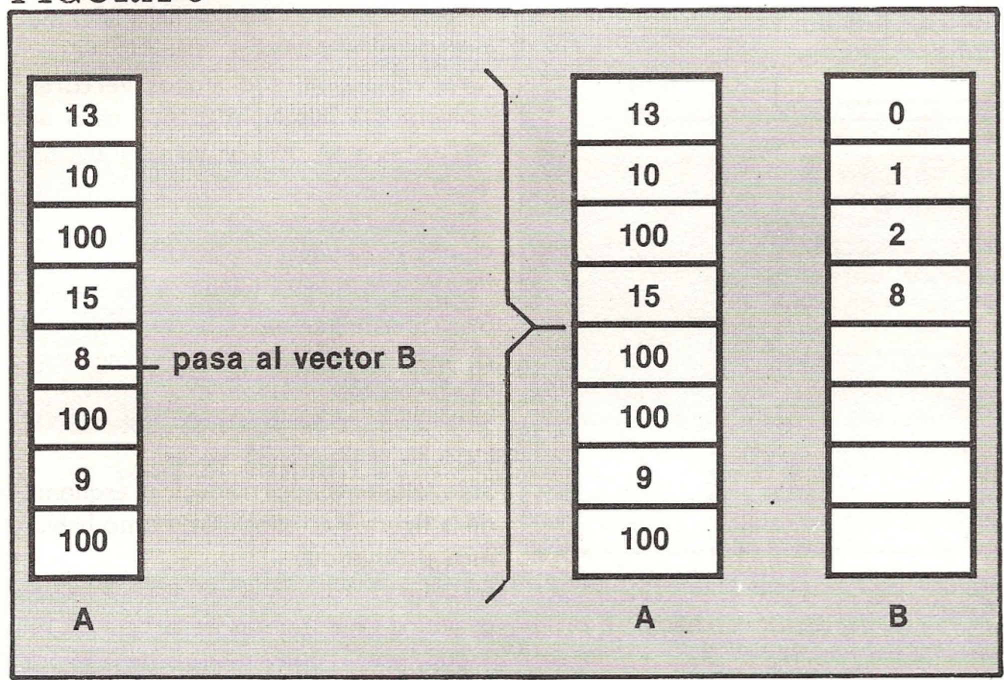


FIGURA 7

A		B
13 (pasa al vector B)	100	0
100	100	1
100	100	2
15	100	8
100	100	9
100	100	10
100	100	13
100	15 (pasa al vector B)	15

FIGURA 8

(K = 1,N) = FOR K =1 TO N

A(J) > A(I) = IF A(J) > A(I)

I ← 1 = I = 1

*- ordenación rápida

Iremos comentando y detallando sólo los dos primeros métodos en esta nota. Aclaramos que si se desea ordenar el vector en forma descendente, el mecanismo es muy similar.

METODOS DE SELECCION I

Este método utiliza dos vectores: el de entrada (donde se encuentran los datos a ordenar) y un segundo vector donde se irán colocando los datos ordenados. Con el solo fin de facilitar la explicación, identificaremos al primer vector con la letra A y al segundo con la B.

Los pasos en que consiste este método son los siguientes:

1 - buscar el menor dato en el vector A y colocarlo en la primera posición o caja del B.

En la posición donde se encontró el menor dato del vector A, se coloca un "tope". Este tope debe ser una cifra mayor a las que se almacenan en el vector A. Si no se conoce de antemano cuál puede ser el mayor dato, se coloca la cifra máxima que admite el sistema.

2 - repetir el procedimiento 1, pero buscando ahora el segundo elemento más pequeño. Una vez hallado se lo ubica en la segunda posición del vector B. En la caja del vector A, donde se encontró el segundo dato más pequeño, se coloca el "tope".

3 - repetir el procedimiento 1, hasta obtener todos los datos del vector A, ordenados según habíamos prefijado.

Realicemos estos pasos con nuestro vector de prueba de la figura 2.

El menor elemento es la cifra 0, ubicada en el casillero 6. Pasemos este dato a la primera posición del vector B y coloquemos un tope, como por ejemplo, la cifra 100.

El resultado lo podemos observar esquemáticamente en la figura 3.

Como vemos, hemos cambiado la sexta posición del vector A y comenzamos a llenar el vector B.

Hagamos entonces una segunda pasada

FIGURA 9

Comienzo

Recorrer el vector desde el primer elemento hasta el último

Suponemos que la posición del menor número es en la caja 1

Recorremos el vector desde el segundo hasta el último elemento

Observamos si el elemento es mayor que alguno de los restantes elementos del vector

NO

Incrementa el ciclo

SI

Guardar el número del casillero donde se halla el menor dato

Almacenar el menor dato del vector A en el vector B

Rellenar el casillero correspondiente del vector A con el número tope

al vector A, ya modificado.

Siempre se trabaja sobre el último vector A modificado.

El segundo elemento más pequeño es la cifra 1 ubicada en la octava posición del A. Lo colocamos en el segundo casillero del B y rellenamos la octava caja del A con el tope 100, obteniendo los vectores como muestra la figura 4.

Con una tercera ejecución de este método, los vectores resultantes serían los graficados en la figura 5.

Con una cuarta pasada, el menor elemento sería la cifra 8 del casillero 5. Los vectores modificados serían los que se ven en la figura 6.

Finalmente, en la séptima pasada, encontramos como menor elemento a la cifra 13 del primer casillero del vector A. Y se termina de ordenar los vectores como se ve en la figura 7.

Notemos que a la séptima pasada, en el vector A sólo quedan dos cifras por ordenar: las cifras "13" y "15".

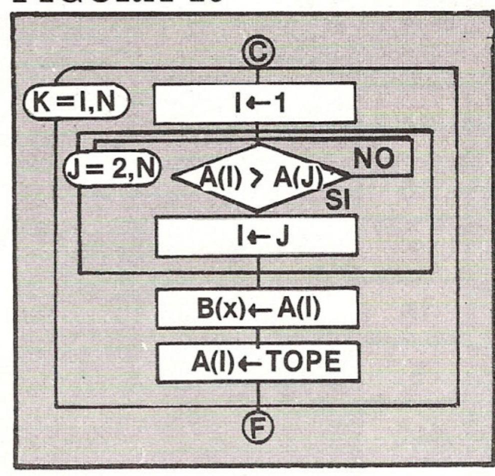
Al finalizar la séptima pasada, se ordenó la cifra menor, es decir el número "13", y sólo queda el "15" sin ordenar. Pero como es el único restante, no hace falta realizar una octava pasada, porque si el procedimiento se desarrollo correctamente, debería ser el menor número que resta ordenar del vector A, ya que los restantes casilleros deberían contener el número "tope".

Aquí se nota por qué buscamos un "tope" que sea realmente un número mayor que cualquiera de los datos almacenados en el vector a ordenar.

Veamos ahora un esquema que simbolice los pasos a realizar por este procedimiento, pero esta vez, expresados con símbolos de computación.

Para los que no están muy familiarizados con estos símbolos, les aclaramos los significados de los mismos en la figura 8.

FIGURA 10



Comentemos el esquema del procedimiento primero con palabras como en la figura 9, y luego con símbolos como realmente lo hacen los programadores.

Como podemos ver, recorremos todo el vector, buscando el casillero donde se halla el menor dato. Una vez encontrado, guardamos el dato de ese casillero en el vector B y colocamos en el casillero del A (donde se hallaba el menor dato) el número "tope".

Una vez modificado ambos vectores, volvemos a buscar el menor dato del vector A y se repiten las operaciones anteriores.

Hay que destacar que este ciclo se hace una vez menos que la cantidad total de elementos almacenados en el vector.

Si en nuestro ejemplo, la cantidad total de cajas es 8, entonces el ciclo se repetirá 7 veces.

Si la cantidad de casilleros fuese 36, el ciclo se realizaría 35 veces.

Sólo falta entonces traducir el esquema de la figura 9 en símbolos, como habíamos prometido.

En la figura 10, vemos cómo se pasa del esquema con palabras de la figura 9, al esquema sólo con símbolos de la figura 10.

Observemos que para este pasaje se utilizaron los símbolos explicados en la figura 8.

Nuestro vector final ya no es el A, sino el vector B con los datos ordenados según habíamos pedido.

Este método es sencillo, pero tiene dos desventajas:

*- se repite el ciclo de buscar el menor elemento varias veces. En consecuencia, hace que el método sea lento si el vector es muy extenso.

*- necesita espacio para un vector auxiliar donde se guardarán los datos ordenados.

Este método es sencillo de implementar si queremos ordenar vectores de pocos casilleros, como por ejemplo, de 10 ó 15 elementos.

HAL-X CARTRIDGES





juegos olimpicos



rally 2000



golf

algunos de nuestros titulos

rally 2000
la ruta delas piramides
el ultimo mohicano
ruta suicida
tenis
pollito
tirabombas
simulador de vuelo
guerra estelar
teatro del misterio

juegos olimpicos 1
ping pong
hal sports 2
hal sports 1
aventuras en el circo
mision en ganimedes
el jardinero del rey
base alfa
aventura en la antartida
y muchos mas......

juegos olimpicos 2
elefante azul
mision suicida
sintetizador musical
camino al futuro
planeador
espionaje alpino
kamikaze
pool
golf
ajedrez



Sociedad Anónima

ADMINISTRACION Y VENTAS
TEL.: 93-3086/87 97-6476

DOS JUEGOS POR CARTRIDGE
GARANTIA POR UN AÑO


TOSHIBA
HX-20
MSX
La Tecnoseducción.

usuaria '87

V Congreso Nacional de Informática, Teleinformática y Telecomunicaciones.

Informática y Comunicaciones: Recursos para la excelencia.

Del 1° al 5 de Junio de 1987. Plaza Hotel.

En el marco de Usuaria '87 se llevará a cabo Unimática '87: Primer Encuentro de Integración entre la Universidad y la Empresa.

Presentación de trabajos:
Los resúmenes de los trabajos a presentar deberán ser remitidos antes del 15-12-86 a Usuaria.

Areas de Interés (No Excluyentes)

1. Gobierno
2. Educación
3. Banca
4. Producción
5. Derecho
6. Cultura y Sociedad
7. Inteligencia artificial
8. América Latina
9. Tecnologías informáticas
10. Tecnologías de telecomunicaciones
11. Pequeña y mediana empresa

Organiza **usuaria**

Asociación Argentina de Usuarios de la Informática y las Comunicaciones.

Rincón 326 (1081) Capital Federal.
T.E. 47-2631/2855

ALPHA

FIGURA 16

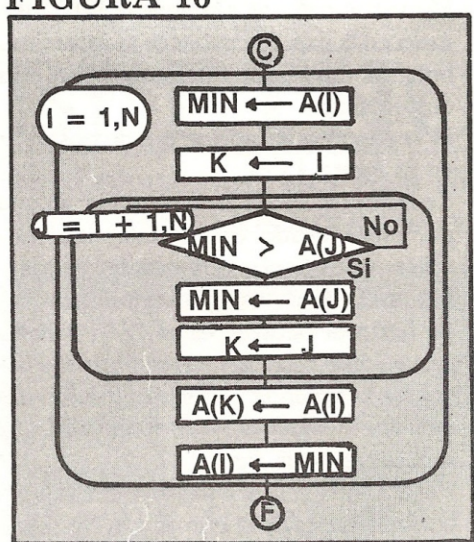


FIGURA 17

```
10 FOR K=1 TO N
20 I=1
30 FOR J=2 TO N
40 IF A(I)>A(J) THEN I=J
50 NEXT J
60 B(K)=A(I)
70 A(I)=TOPE
80 NEXT K
```

FIGURA 18

```
10 FOR I=1 TO N
20 MIN=A(I)
30 K=I
40 FOR J=I+1 TO N
50 IF MIN>A(J) THEN MIN=A(J):K=J
60 NEXT J
70 A(K)=A(I)
80 A(I)=MIN
90 NEXT I
```

Veamos ahora el esquema de este método, primero con palabras (ver figura 15).

Podemos observar en este esquema, que primero se busca el menor dato almacenado desde la primera caja en adelante del vector.

Una vez hallado, se intercambian los datos entre la primera caja y el casillero donde se encontró el mínimo valor.

Luego se busca el menor dato a partir la segunda caja y una vez encontrado se realizan las mismas operaciones.

Notemos que, con cada ciclo, disminuye la cantidad de casilleros para buscar el menor dato.

El vector se va ordenando desde arriba hacia abajo, como pudimos ver en el ejemplo anterior.

Pasemos este esquema con palabras, al esquema con sólo símbolos de computación. En la figura 16 se puede observar el resultado de este pasaje de símbolos.

Este método es más eficiente que el de selección I para vectores de mayor dimensión, como por ejemplo de 20 ó 30 casilleros.

En la figura 17 y 18 se encuentra el listado de los métodos de selección I y II respectivamente.

Sólo se tradujo en sentencias el esquema de símbolos comentado más arriba y esquematizados en las figuras 10 y 16

Pero también se emplean otros métodos (mencionados al comienzo de esta nota). Cada método tiene su ventaja y su desventaja. No existe una regla que indique cuándo, exactamente, se deba aplicar un método en particular. Esto sólo va a criterio del programador y también depende de las características del vector y del sistema con el cual se está operando.

FIGURA 11

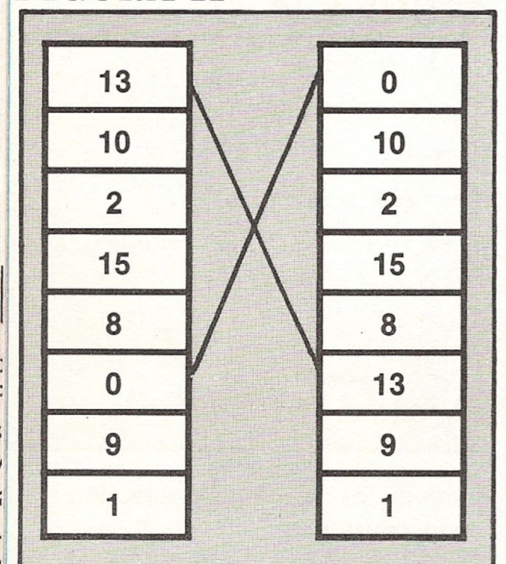


FIGURA 12

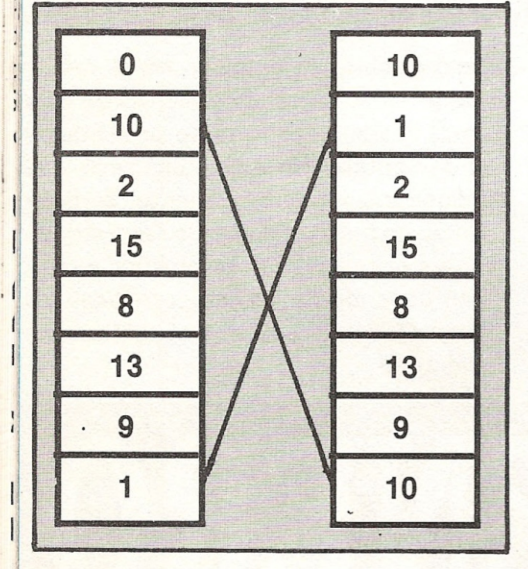


FIGURA 13

0		0
1		1
2	Pasa a la 3ª caja. Entonces el vector antes, no se producen cambios en la tercer pasada	2
15		15
8		8
13		13
9		9
10		10

FIGURA 14

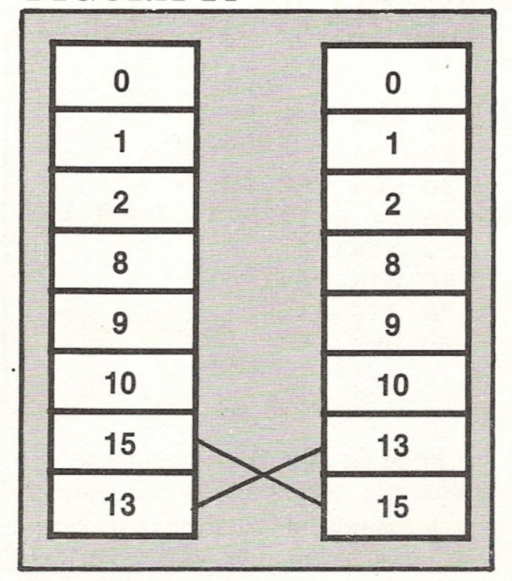
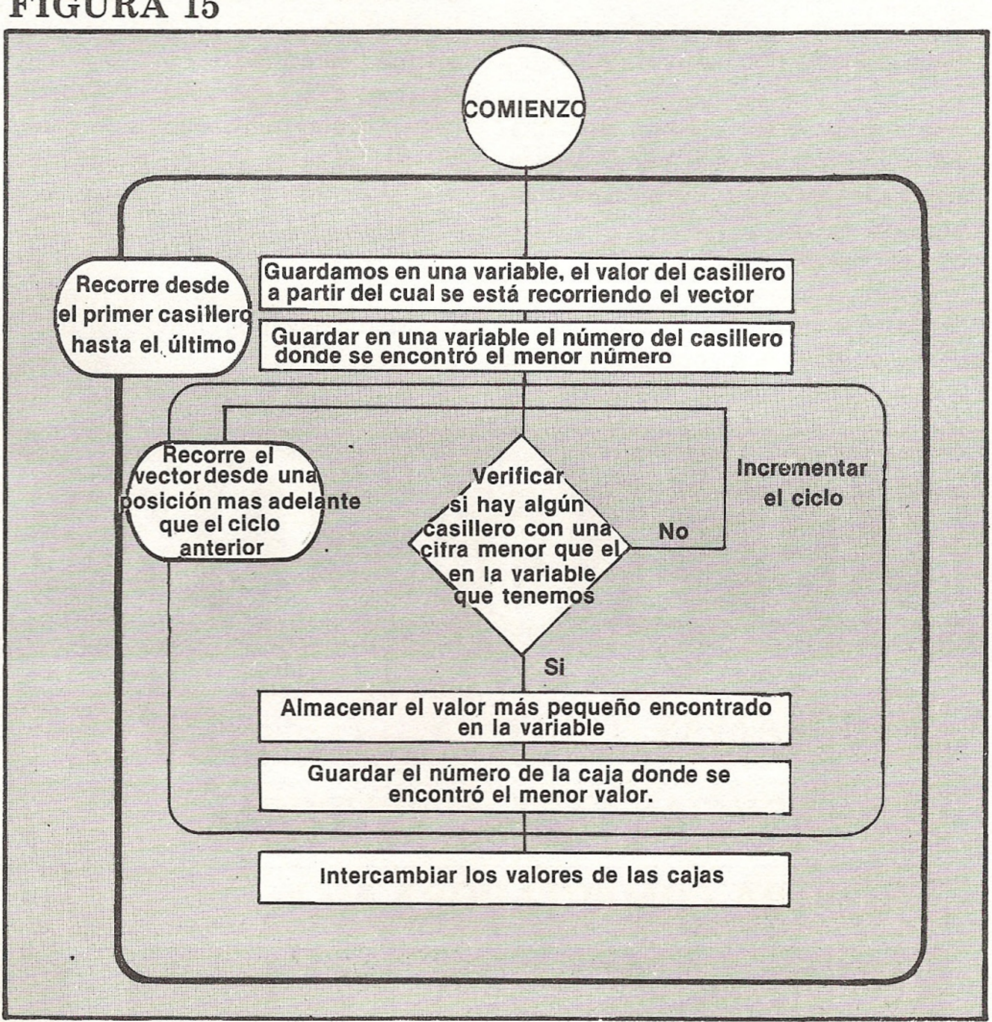


FIGURA 15



METODO DE SELECCION II

Este método sólo utiliza un vector, y no necesitamos buscar un número "tope".

Sus pasos son:

1 - seleccionar el elemento más pequeño del vector.

2 - intercambiarlo con el elemento ubicado en la primera caja del mismo vector.

3 - buscar a partir del segundo casillero en adelante, el dato más pequeño.

4 - intercambiarlo con el valor almacenado en la segunda caja.

5 - repetir este procedimiento con los restantes casilleros.

Con este método, el vector ordenado sera el mismo donde se encontraban inicialmente los datos. Los valores no se pasan a otro vector auxiliar.

Tomemos el mismo vector de la figura 2, y apliquémosle este método.

El menor dato es el número "0", ubicado en la caja 6. Entonces se intercambian los contenidos de la primera caja con la sexta.

El vector modificado se puede observar en la figura 11.

Realizamos ahora una segunda pasada por este método. En consecuencia, modificaremos la segunda caja del vector. El menor dato desde el segundo casillero hasta el octavo, es el "1", hallado en la última caja. Se intercambian los datos de la segunda y octava posición del vector, quedando éste modificado como se ve en la figura 12.

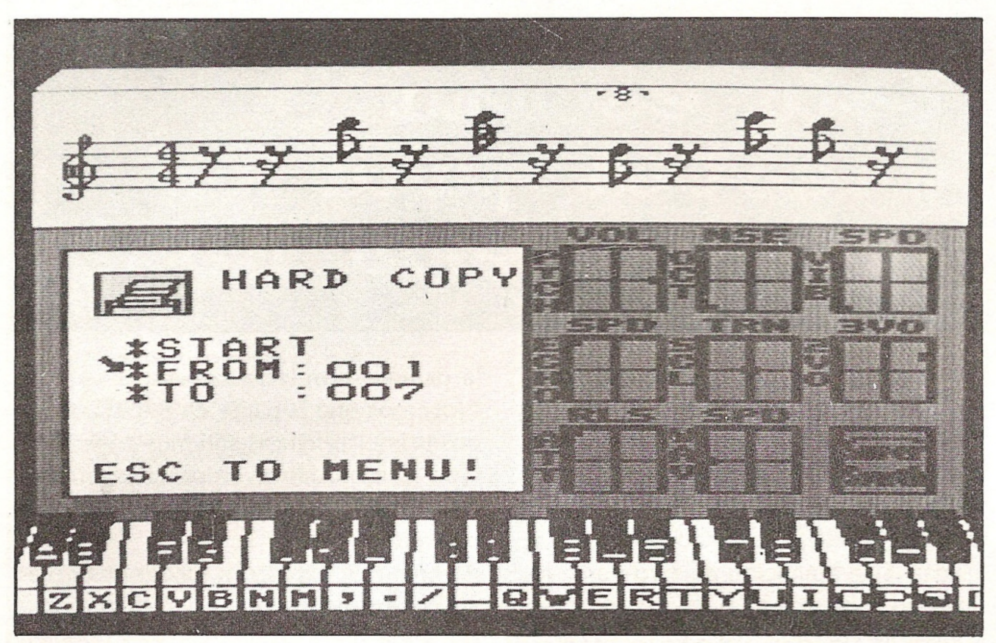
Pero con la tercera pasada, el casillero a modificar sería el número 3. Se busca el menor dato a partir de la tercera caja, como se puede notar es el número "2" ubicado en la tercera caja.

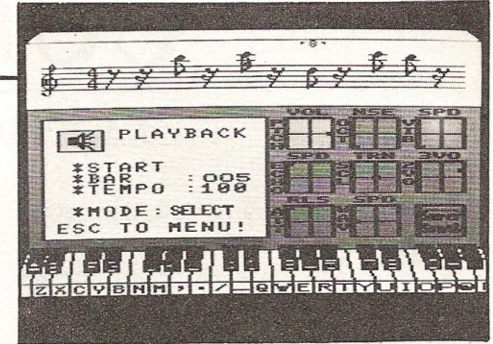
En este paso, da la casualidad de que no se producen cambios, como se puede observar en la figura 13.

Continuando con los mismos pasos, obtenemos finalmente en la séptima pasada que el menor número entre el casillero séptimo y el octavo es el número "13" de la última caja. Por lo tanto, el número 15 pasará al octavo lugar y al 13 el séptimo como se esquematiza en la figura 14.

COMPONIENDO COMO BEETHOVEN

Este es un soft del tipo de los utilitarios, que les da una ayuda a los amantes compositores de la música.





Hace varios años atrás, comenzaron a aparecer en los negocios especializados en música los primeros sintetizadores.

Muchos negaban el triunfo de estos nuevos aparatos que utilizaban por primera vez un chip.

Pero rápidamente se fueron mejorando estos órganos electrónicos y comenzaron a invadir el mercado.

Y no transcurrió demasiado tiempo para poder encontrar sintetizadores o software musical, pero esta vez comandado desde una computadora hogareña.

"Super Sinth" es uno de los mejores utilitarios que le permite a los usuarios de las computadoras MSX, poder crear temas musicales sin necesidad de ser expertos músicos.

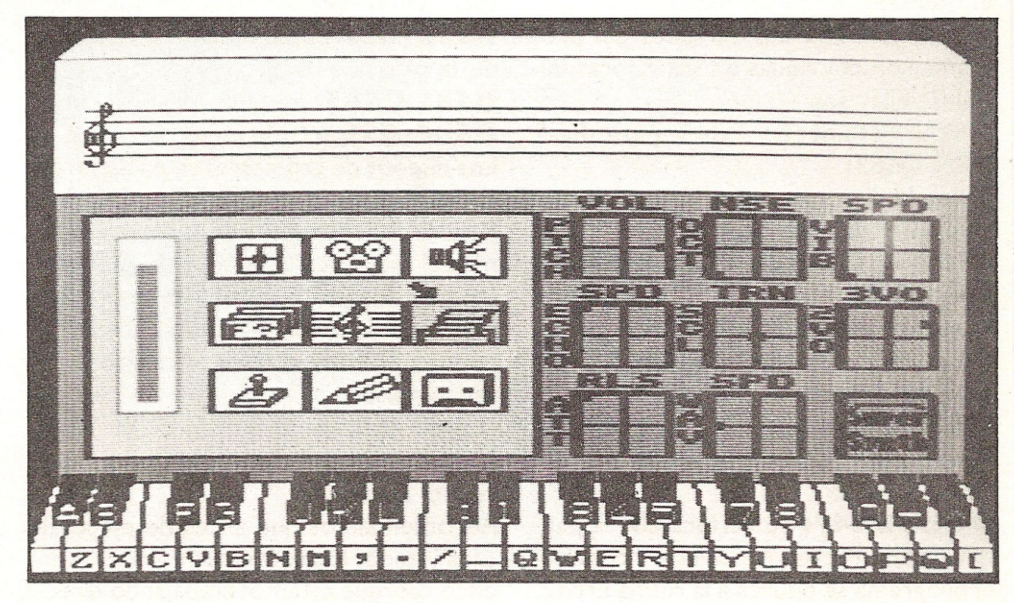
Sólo hay que tener buen oído para componer, y muchas ganas de alegrar un poco las cosas con nuestros sonidos.

No será demasiado el tiempo que tardemos en dominar prácticamente todas las opciones que nos ofrece este poderoso soft-sonoro.

Merece resaltarse la simplicidad que ofrece para crear los más elaborados efectos sonoros y musicales. Muchas son las opciones y todas muy fáciles de reconocer y utilizar.

LAS OPCIONES

A primera vista nos veremos sentados frente a una especie de piano de pared, que en su parte media nos muestra el panel de una computadora musical y en la parte superior, un utilísimo pentagrama para todos aquéllos que saben interpretarlo o leerlo.



El panel de computadora al que hacíamos referencia está dividido en dos partes: la media izquierda es la destinada a los distintos menúes, y la derecha, a los ítems que hacen que nuestro computarizado órgano suene como suena.

Esto significa que podremos cambiar el volumen, ruido, forma de onda, el ataque, release, velocidad del eco, vibrato, etcétera. Todo esto será posible si elegimos la opción de la cruz del menú principal.

De otra forma, si no nos animamos aún a tocar estas, a veces, sutiles opciones, podremos obtener 10 tipos de resultados musicales distintos

Así cada vez que pulsemos una tecla de función, observaremos cómo se configura en base a los ítems anteriores, el tipo de sonido elegido.

Podremos ver lo que tocamos en el pentagrama, grabar la música interpretada mientras nos marca el tiempo del compás elegido. Luego también podremos verlo en el pentagrama, con su correspondiente clave, etcétera.

Otra de las asombrosas opciones es la de poder sacar una copia de la partitura en una impresora, que como de costumbre ha de ser MSX.

Si es nuestro gusto podremos manipular, directamente sobre el pentagrama, las notas y silencios para retocar los efectos no deseados, producto de nuestro apuro por interpretar correctamente una melodía mientras la grabamos.

Recordemos por último que la grabación se efectúa en RAM, pero que luego podremos pasar a cassette, para recordar nuestros mejores momentos.

CURSIVA

CLASE: UTILITARIO *AUTOR: DANIEL GALIMBERTI*

Este programa escribe letras cursivas definidas sobre una matriz de 8 de ancho por 16 de alto. Es un programa utilitario de tipo gráfico. Puede ser utilizado en la programación de otros software que lo contenga, como por ejemplo programas destinados a niños, por la claridad y la fácil lectura de los caracteres cursivos. El juego de caracteres incluye a todas las letras del abecedario, (incluidas la eñe y las vocales acentuadas) tanto en mayúsculas como en minuscúlas, así como los corchetes, paréntesis, símbolos aritméticos, comillas, números, potencias, subrayado, puntos, comas, etcétera.

Esta es una muestra para la revista Load MSX!!

BLOQUES DEL PROGRAMA

D400-D418: Son leídos los argumentos que se encuentran en la función USR. En base a éstos, entre las direcciones D418-D437 es calculada la dirección de la VRAM en la que será escrito el mensaje.

D438-D446: Las variables del programa son colocadas en sus valores iniciales. Ellas son:

PPAN (Puntero de pantalla-2 bytes-D583)

Posee la dirección de la Vram a la cual es tranferida cada byte de los diseños.

CTEX (Contador de texto-1 byte-D585)

Posee la cantidad de caracteres que faltan escribir y es la señal para la finalización del programa.

PTEX (Puntero de texto-2 bytes-D581): Posee la dirección del próximo byte del texto a leer.

D44A-D44F: Se verifica que la pantalla está en Screen 2. Si esto no es así, el programa se bifurca a la rutina Erro2.

D452-D4B1 Se calcula la dirección de la RAM en la que se encuentra el diseño es cargada en HL. En caso de estar ser un caracter no ordenado secuencialmente, el programa se bifurca a una rutina específica para ese caracter (p.ej. AACENT) donde la dirección del diseño es cargada en, HL. En caso de estar ordenado secuencialmente, entre las direcciones D49C y D4B1 se determina la dirección. Si el caracter no está definido, el programa salta a la rutina ERROR.

D4B2-D4C7: Los diseños son transferidos a la pantalla.

D4CA-D4E8: Las variables son actualizadas. Si la ejecución del programa ha finalizado, bifurca a VOLVER, en caso contrario vuelve a CICLO (D452).

D4EB-D525: Determinación de las direcciones de los diseños de los caracteres.

D526-D54F: Rutina de error por caracter no definido.

D552-D578: Rutina de error por no estar en Screen 2.

D57B-D580: Vuelta al Basic, ajuste de la pila.

D581-D585: Variables del programa.

D586-D5BA: Mensajes de error.

Los diseños de caracteres se encuentran entre las direcciones CC60 y D240.

Para copiar este utilitario, primero tipiemos el listado de la figura 1, lo ejecutamos, y luego entramos la sentencia NEW para borrar el listado BASIC y quedarnos con los códigos cargados en la memoria.

Luego copiemos el listado de la figura 2 y procedamos de la misma manera que con el anterior.

Recomendamos que antes de ejecutar cada listado, lo guardemos en cinta o en disco, porque estamos trabajando directamente con códigos sobre la RAM.

Si copiamos algún código erróneo de las sentencias DATA, saldrá un mensaje de error y el número de línea donde se copió el dato equivocado.

Una vez cargados correctamente los dos listados de la figura 1 y el 2, copiemos el programa cargador de la figura 3.

Al ejecutar este pequeño programa, veremos por fin, cómo se escribe sobre la pantalla con letra cursiva.

Notemos que la parte en BASIC de las dos primeras figuras son muy parecidas. Con sólo modificar un par de números de la primera, ya tendrás parcialmente copiado el listado 2.

Al ejecutar estos dos programas (recuerda primero copias uno y lo ejecutas, y luego copias el segundo y también lo haces correr), no apreciarás ningún cambio que represente la función del programa que se está ejecutando, salvo un mensaje de "espere..."

Recién al cargar en memoria el listado de la figura 3, notarás cómo funciona el programa.

Para cambiar el mensaje que quieres que se imprima en cursiva, deberás cambiar el contenido entre comillas de la línea.

Si deseas utilizar este programa como una rutina de otro algoritmo, recuerda que el programa se ejecuta haciendo un **def usr = &hd400** y un **usr("texto"),x,y** en screen 2. Si la pantalla no se encuentra en screen 2 en el momento de ser ejecutado el programa, o si el caracter usado como argumento no está entre los definidos por el programa, se generarán mensajes de error. De la misma manera, si no se incluyen los argumentos x e y en la función usr, el programa generará un Syntax Error.

X e Y son coordenadas de pantalla, y para determinar sus valores hay que calcularlos como para cualquier función gráfica en screen 2. Los valores serán modificados para ubicar a las letras en coordenadas que sean múltiplos de ocho.

El largo máximo de una frase es el que acepta la función USR como argumento: 255 caracteres.

Los argumentos pueden ser escritos directamente o por medio de variables.

Figura 1

```
10 CLS:KEY OFF:LOCATE 10,9:PRINT
"Esperá un momento ..."
20 D=1000:SUMA=0
30 FOR F=&HCC60TO &HD211 STEP 47
40 FOR I=0 TO 46:READ A$
50 SUMA=SUMA+VAL("&H"+A$)
60 POKE I+F,VAL("&H"+A$)
70 NEXT I:READ B$
80 IF SUMA <>VAL("&H"+B$) THEN P
RINT"error en el data ";D:STOP
90 D=D+10:SUMA=0:NEXT F
1000 DATA 0,0,1C,1C,38,60,0,0,0,
0,0,0,0,0,0,0,4,8,10,10,20,20,20,
20,20,20,20,20,10,10,8,4,20,10,8,
8,4,4,4,4,4,4,4,4,8,8,10,2A8
1010 DATA 20,0,0,0,0,0,82,44,28,
10,28,44,82,0,0,0,0,0,0,0,0,10,
10,10,FE,10,10,10,0,0,0,0,0,0,0
,0,0,0,0,0,1C,1C,38,60,0,43A
1020 DATA 0,0,0,0,0,0,0,0,0,FE
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
1C,1C,1C,0,0,0,0,0,0,0,1,2,4,8,
10,20,40,80,0,251
1030 DATA 0,0,0,0,0,0,38,44,82,8
2,82,82,82,44,38,0,0,0,0,0,0,8,
18,28,8,8,8,8,8,1C,0,0,0,0,0,0,
38,44,44,8,10,20,40,74,4C,606
1040 DATA 0,0,0,0,0,0,38,44,4,
18,4,2,2,44,38,0,0,0,0,0,0,10,10,
12,22,22,7F,2,2,2,2,0,0,0,0,0,0
,7C,40,40,78,44,2,2,44,419
1050 DATA 38,0,0,0,0,0,0,1C,20
,20,40,4E,51,61,22,1C,0,0,0,0,0,0
,0,62,5C,4,8,3C,10,20,20,20,0,0,0
,0,0,0,0,18,24,24,18,24,42,42,4A8
1060 DATA 24,18,0,0,0,0,0,0,1C
,22,22,1E,2,2,4,8,10,0,0,0,0,0,0,
0,0,0,0,30,30,0,0,30,30,0,0,0,0,0
,0,0,0,0,0,18,18,0,1CA
1070 DATA 0,18,18,30,0,0,0,0,0
,0,2,4,8,10,20,10,8,4,2,0,0,0,0,0
,0,0,0,7E,0,0,7E,0,0,0,0,0,0,0,
0,0,20,10,8,4,2,1F6
1080 DATA 4,8,10,20,0,0,0,0,0,
0,38,44,42,2,4,8,10,10,0,18,18,0,
0,2,5,5,9,29,59,F,51,91,,91,A1,41,
0,0,0,0,30,68,A8,A8,70,28,24,6F7
1090 DATA 22,2A,B2,B4,48,0,0,0,0
,4C,92,A2,A4,48,70,40,40,44,4A,2C
,18,8,10,0,0,4,38,4C,4A,2A,9,11,1
1,11,71,92,6C,0,0,0,0,48,94,78,10
,18,20,A83
1100 DATA 40,40,44,4A,2C,18,8,10
,0,0,31,4A,94,90,50,3A,54,10,50,9
0,90,60,0,0,0,0,6C,92,A4,79,22,1E
,4,4,22,52,42,3C,0,0,0,0,A2,E5,A5
,25,26,B93
1110 DATA 3C,64,A4,24,24,A4,46,0
,0,0,0,41,62,5E,2,2,2,32,49,49,41
,22,1C,0,0,0,0,8,8,18,28,48,8,A,C
,18,28,48,88,88,88,50,20,A3,E4,A4
,24,A55
1120 DATA 28,30,28,24,64,A4,A4,4
3,0,0,0,0,88,94,54,38,10,10,10,10
,16,71,99,66,0,0,0,0,44,44,6C,6C,
6C,54,54,54,44,46,45,82,0,0,0,0,6
1,22,22,ABF
1130 DATA 32,32,2A,2A,26,26,A2,A
2,42,0,0,0,0,2C,52,52,A1,A1,A1,81
,81,81,42,42,3C,0,0,0,0,1C,62,A1,
21,61,32,2C,20,20,A0,A0,40,0,0,0,
0,2C,52,BBD
1140 DATA 52,A1,A1,A1,81,81,81,5
A,46,3E,1,0,0,0,18,64,A2,22,64,38
,24,22,22,A2,A2,41,0,0,0,0,8,C,14
,12,22,41,81,1,31,49,22,1C,0,0,0,
0,41,A78
1150 DATA 42,7E,84,8,7C,20,4E,91
,89,42,3C,0,0,0,0,60,92,92,92,92,
92,92,12,12,12,12,D,0,0,0,0,40,A0
,A4,A7,24,24,24,24,24,24,24,18,0,
0,0,0,B2E
1160 DATA 80,40,54,57,54,54,54,5
4,54,54,54,28,0,0,0,0,81,41,41,22
,14,8,14,22,41,41,41,81,0,0,0,0,0
,60,94,94,94,94,94,94,D,6,4,C,14,
24,24,AF1
1170 DATA 18,81,C2,BC,4,8,D0,24,
58,80,BD,C3,81,0,0,0,0,3C,20,20,2
0,40,80,40,20,20,20,20,3C,0,0,0,0
,0,0,0,0,80,40,20,10,8,4,2,1,0,0,
947
1180 DATA 0,0,78,8,8,8,4,2,4,8,8
,8,8,78,0,0,0,0,10,28,44,82,0,0
,0,0,0,0,0,0,0,0,0,FF,FF,0,0,0,
0,0,0,0,0,0,0,42E
1190 DATA 0,0,0,0,1C,1C,E,3,0,0,
0,0,0,0,0,0,0,0,0,0,0,34,4C,4
4,44,C4,45,4D,36,0,0,0,0,30,48,48
,48,48,50,60,40,C8,4F,48,30,6AC
1200 DATA 0,0,0,0,0,0,0,1C,22,
42,44,C0,41,22,1C,0,0,0,0,0,4,4,4
,34,4C,44,44,C4,45,4D,32,0,0,0,0,
0,0,0,0,1C,22,42,4C,F0,41,22,6BE
1210 DATA 1C,0,0,0,0,10,28,28,28
,28,28,28,30,F0,2B,3C,24,24,24,24
,18,0,0,0,0,34,4C,44,44,C4,4D,56,
24,C,14,24,18,20,50,50,50,50,50,6
0,40,D0,69,9C3
1220 DATA 4A,44,0,0,0,0,0,0,18
,18,0,20,60,A0,21,22,1C,0,0,0,0,0
,0,18,18,0,10,30,50,90,17,18,30,5
0,90,90,60,18,24,24,24,28,30,20,6
8,B4,7D4
1230 DATA 39,25,22,0,0,0,0,18,24
,24,24,28,30,20,60,A0,21,22,1C,0,
0,0,0,0,0,0,0,0,54,AA,AA,AA,2B,2B
,2A,0,0,0,0,0,0,0,0,28,54,94,6B
D
1240 DATA 94,15,15,12,0,0,0,0,0,
0,0,0,0,1C,32,6A,A7,22,22,1C,0,0,
0,0,0,0,0,0,10,18,34,54,94,15,15,
12,10,10,10,10,0,0,0,0,34,4C,4C
F
1250 DATA 44,C4,45,4D,35,6,C,14,
4,0,0,0,0,40,68,50,90,11,9,6,0,
0,0,0,0,0,0,0,8,18,24,C2,3,12,C
,0,0,0,0,10,10,7E,10,10,10,596
1260 DATA 10,30,D0,11,12,C,0,0,0
,0,0,0,0,0,48,48,48,C8,49,5A,24
,0,0,0,0,0,0,0,0,40,A4,A4,A7,24
,24,18,0,0,0,0,0,0,0,4A,67F
1270 DATA AA,AA,AA,AA,2B,2A,14,0
,0,0,0,0,0,0,22,52,52,54,48,95,
25,22,0,0,0,0,0,0,0,44,A4,A4,A4
,A4,2D,16,4,C,14,24,18,0,0,0,0,8C
6
1280 DATA 30,48,88,88,90,29,6,C,
14,24,24,18,0,8,10,0,34,4C,44,44,
C4,45,4D,36,0,0,0,0,4,8,0,18,24
,44,48,F0,41,22,1C,0,0,0,0,0,0,4,
7BA
1290 DATA 8,0,10,10,30,D0,11,9,6
,0,0,0,0,0,4,8,0,18,24,64,5F,C4,4
4,44,38,0,0,0,0,0,8,10,0,48,48,
48,C8,49,59,26,0,0,0,0,0,0,657
1300 DATA 34,48,0,28,54,94,94,15
,15,12,0,0,0,0,32,4C,81,42,42,62,
52,4A,46,42,42,81,0,0,0,0,0,0,C,C
,0,8,8,10,20,40,40,40,22,1C,0,0,0
,77E
1310 DATA 0,0,18,18,0,18,18,18,1
8,18,18,18,18,18,18,18,18,18,18,1
8,18,18,18,18,18,0,18,18,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,240
```

Figura 2

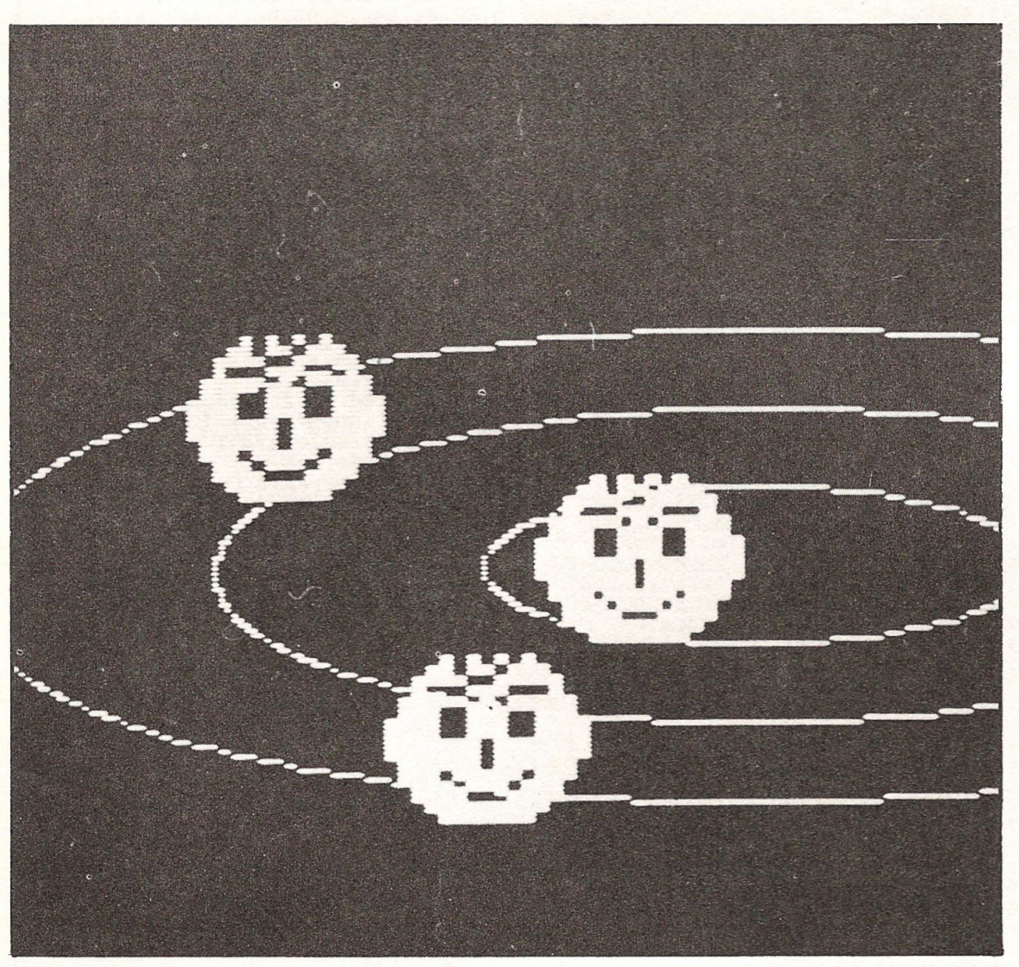
```
10 CLS:KEY OFF:LOCATE 10,9:PRINT
"Esperá un momento ..."
20 D=1000:SUMA=0
30 FOR F=&HD400 TO &HD572 STEP 3
7
40 FOR I=0 TO 36:READ A$
50 SUMA=SUMA+VAL("&H"+A$)
60 POKE I+F,VAL("&H"+A$)
70 NEXT I:READ B$
80 IF SUMA <>VAL("&H"+B$) THEN P
RINT"error en el data ";D:STOP
90 D=D+10:SUMA=0:NEXT F
1000 DATA E5,2A,F8,F7,22,81,D5,E
1,EB,C1,E1,E5,C5,D5,CF,2C,CD,B,40
,E3,C5,D5,C1,D1,69,26,0,6,3,CB,3D
,10,FC,6,8,CB,25,142F
1010 DATA CB,14,10,FA,6,3,CB,3B,
10,FC,6,3,CB,23,10,FC,16,0,19,22,
83,D5,2A,81,D5,7E,32,85,D5,23,5E,
23,56,ED,53,81,D5,ECA
1020 DATA 3A,AF,FC,FE,2,C2,52,D5
,2A,81,D5,7E,FE,20,CA,ED,D4,FE,A0
,CA,F2,D4,FE,82,CA,F7,D4,FE,A1,CA
,FC,D4,FE,A2,CA,1,D5,1A31
1030 DATA FE,A3,CA,6,D5,FE,A4,CA
,B,D5,FE,A5,CA,10,D5,FE,AD,CA,15,
D5,FE,21,CA,1A,D5,FE,A8,CA,1F,D5,
FE,40,CA,26,D5,FE,27,17A7
1040 DATA DA,26,D5,FE,7B,D2,26,D
5,FE,41,D4,24,D5,D6,27,6F,26,0,6,
4,CB,25,CB,14,10,FA,11,60,CC,19,E
5,ED,5B,83,D5,1,8,1180
1050 DATA 0,CD,5C,0,ED,5B,83,D5,
14,1,8,0,E1,9,CD,5C,0,2A,83,D5,1,
8,0,9,3E,7,BD,D4,EB,D4,22,83,D5,2
A,81,D5,23,e3F
1060 DATA 22,81,D5,21,85,D5,35,C
2,52,D4,C3,7B,D5,24,C9,21,30,D2,1
8,C0,21,90,D1,18,BB,21,A0,D1,18,B
6,21,B0,D1,18,B1,21,C0,120C
1070 DATA D1,18,AC,21,D0,D1,18,A
7,21,E0,D1,18,A2,21,F0,D1,18,9D,2
1,10,D2,18,98,21,20,D2,18,93,21,0
,D2,18,8E,3D,C9,32,A1,101C
1080 DATA D5,3A,B0,FC,32,AF,FC,C
D,5F,0,21,86,D5,1,1C,0,11,0,0,CD,
5C,0,21,86,D5,1,1C,0,11,0,18,CD,5
C,0,3E,4,32,BF6
1090 DATA DC,F3,C3,7B,D5,3A,B0,F
C,32,AF,FC,CD,5F,0,21,A2,D5,1,1B,
0,11,0,0,CD,5C,0,21,A2,D5,1,1B,0,
11,0,18,CD,5C,ec5
1100 DATA 0,3E,4,32,DC,F3,C3,7B,
D5,E1,C1,D1,E5,C5,C9,0,0,0,0,0,45
,52,52,4F,52,3A,20,43,41,52,41,43
,54,45,52,20,4E,DD3
1110 DATA 4F,20,44,45,46,49,4E,4
9,44,4F,0,45,52,52,4F,52,3A,20,4E
,4F,20,45,53,54,4F,59,20,45,4E,20
,53,43,52,45,45,4E,20,32,999
```

Figura 3

```
10 COLOR 11,1,15
20 SCREEN 2
30 DEF USR=&HD400
40 PRINT USR("Miguel Angel ya no
 es tan angel como parece, no?"),
7,100
50 GOTO 50
```

MANEJANDO LAS PANTALLAS

Como vimos en el número anterior de esta publicación, el modo 1 permite manejar sprites. En el número anterior hemos visto las tablas de nombres, patrones y de colores para el modo 1.



Existen sentencias en Basic que permiten trabajar fácilmente con este tipo de figuras animadas, pero aquí veremos como podemos manejarlas cambiando algunos datos de lugares estratégicos de la VRAM (memoria de video).

TABLA DE PATRONES PARA SPRITES

La tabla de patrones de sprites tiene una longitud de 2.048 bytes dividida en 256 bloques de 8 bytes cada uno. El comienzo de la tabla está dada por BASE (9).

Es aquí donde se guardan las definiciones de nuestros sprites.

El sprite del plano 1 está definido en los bytes 8 al 15, el sprite del segundo plano, en los bytes 16 al 24 y así sucesivamente.

Pero para definir sprites de 16 x 16 utilizaremos 32 bytes. El sprite de 16x16 del plano 0 está definido en los primeros 32 bytes de esta tabla.

En el esquema de la figura 8, verás la forma en que se guarda la información de los diseños de los sprites para los dos tamaños, 8x8 y 16x16.

Con el listado de la figura 9 vemos cómo podemos definir un sprite sin usar la sentencia SPRITE$. Tal vez esto te parezca un poco más complicado que definirlos con SPRITE$, pero esta forma, con VPOKE, es útil para los programadores en lenguaje de máquina. Todas las direcciones de comienzo de estas tablas son modificables como vimos antes.

Figura 8

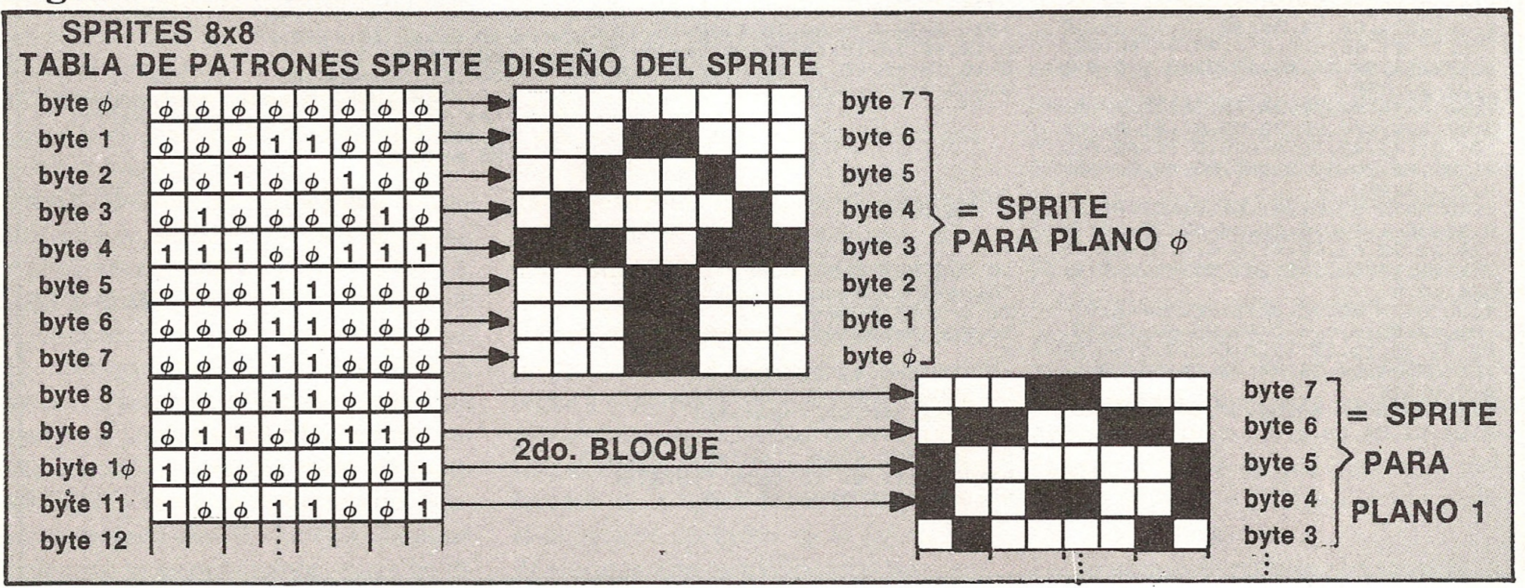


Figura 8

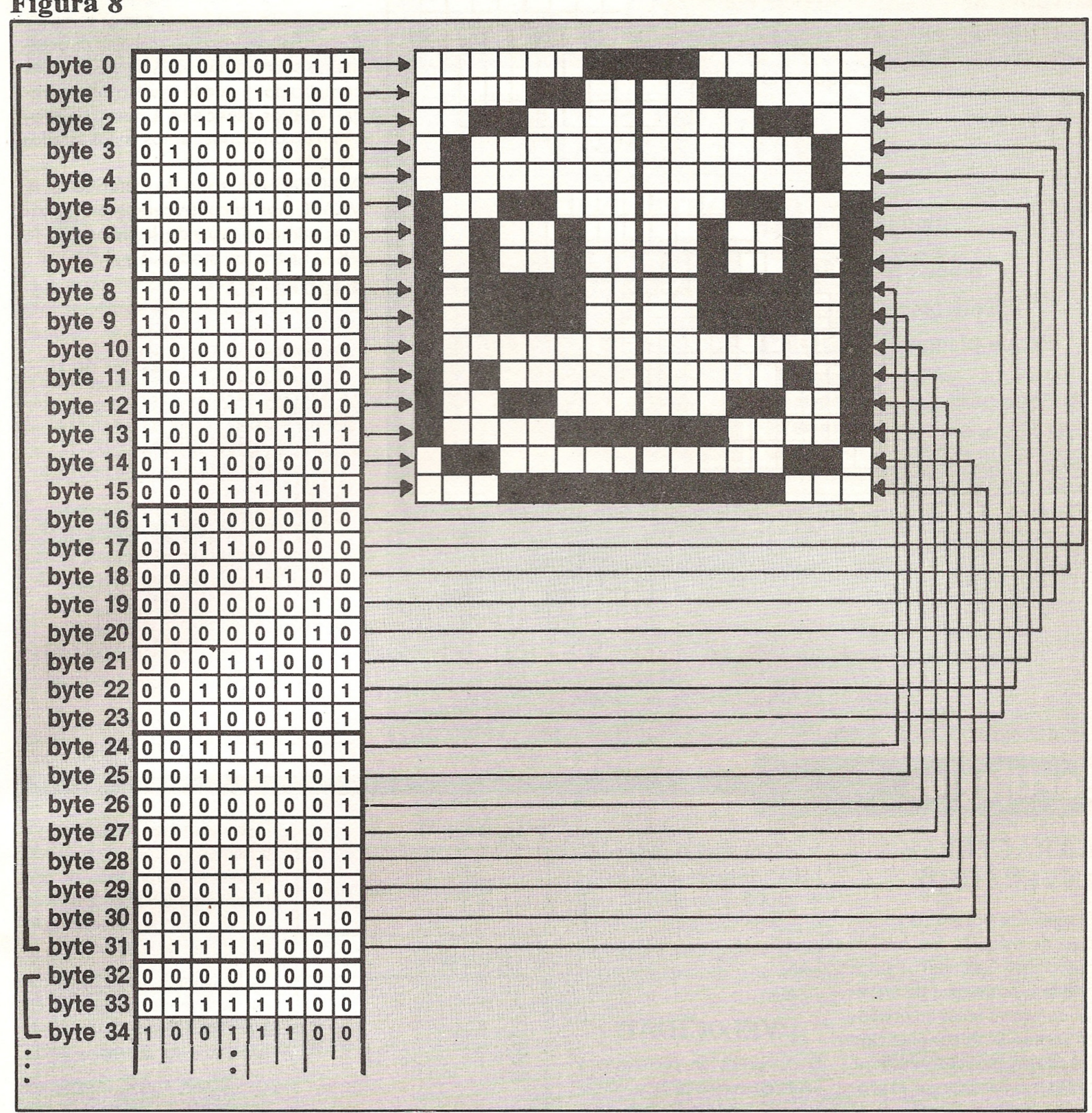


TABLA DE ATRIBUTOS PARA SPRITES

En esta tabla se encuentra almacenada la información correspondiente a cada uno de los planos de sprites. El comienzo de esta tabla está dada por BASE (8). Se encuentra estructurada en 32 tablas, una por plano, y cada una tiene una longitud de 4 bytes (ver figura 10).

Por lo tanto, la longitud total de la tabla de atributos es de 128 bytes (32*4).

Por último, faltaría aclarar la función de cada byte de los bloques en que se divide la tabla de atributos sprites.

Como vemos en la figura 11, el byte 0 guarda la posición de la coordenada X del sprite. Mientras que el byte 1, almacena la posición de la coordenada Y. En el byte 2 se encuentra el número del sprite a visualizar en el plano correspondiente, y por último el byte 3 guarda el color con que se verá al sprite.

Figura 9

```
10 SCREEN 1
20 CO=BASE(9):REM Base (9) da el
 comienzo de ;la tabla de patrone
s sprites para el modo 1.
30 FOR F=CO TO CO+7:REM Abarca 1
os 8 primeros bytes de la tabla
40 VPOKE F,195:REM Definimos el
sprite para el plano 0 por haber
elegido los primeros 8 bytes de 1
a tabla
50 NEXT
60 PUT SPRITE 0,(110,100),6,0:RE
M Colocamos el sprite en pantalla
```

Pero para esto último, sólo se emplean los cuatro bits más bajos del byte 3. Los bits 6, 5 y 4 no tienen uso aún. En cambio el bit 7 es el "bit de anticipación de reloj". Esto significa que al colocar este bit en 0, la esquina superior izquierda se posicionará en el punto X, Y, en cambio si es 1, se ubicará en el punto X-32, Y.

Finalmente, en la figura 12 veremos cómo utilizar esta tabla.

Para todos estos temas avanzados, es aconsejable practicar cambiando los valores a los programas propuestos y probar inventar los propios.

De esta manera advertirás cambios que

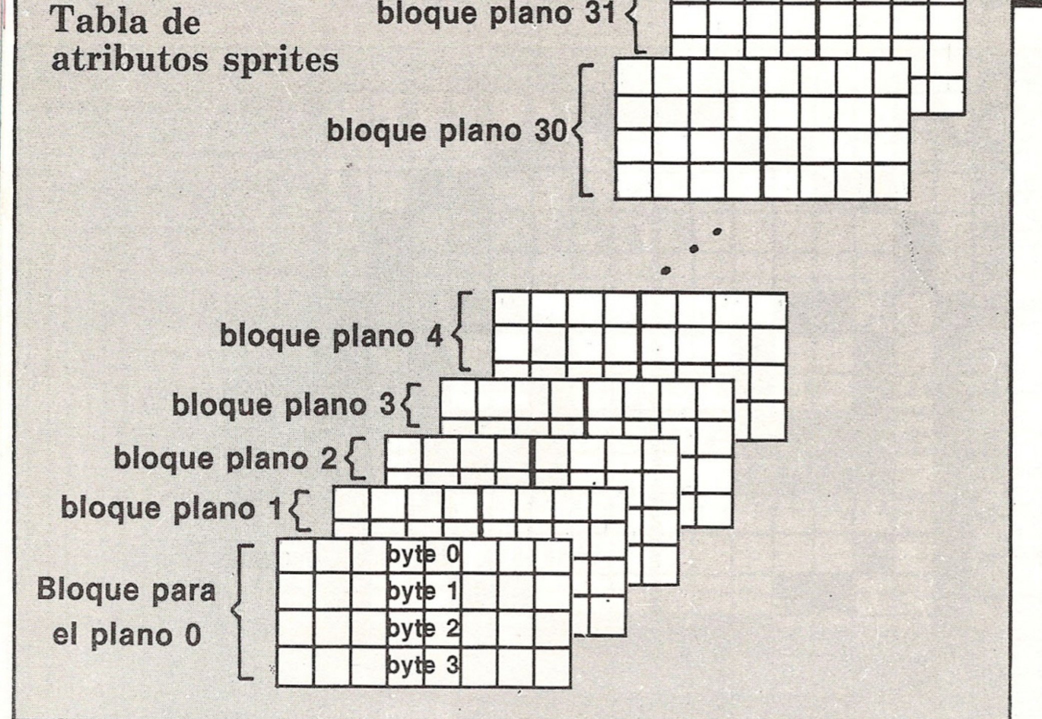
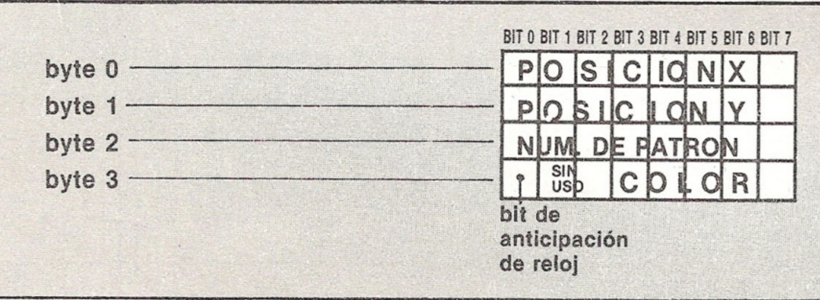


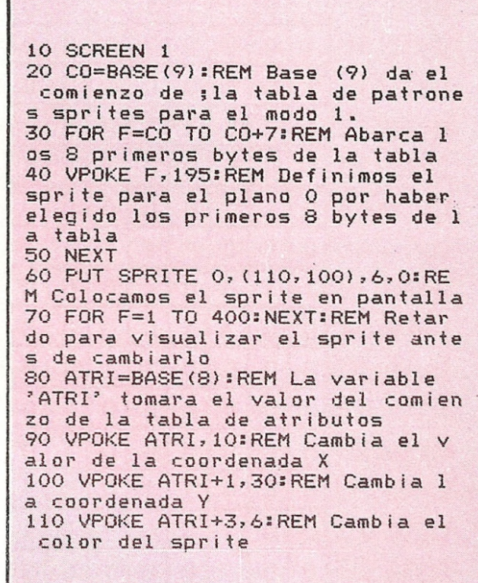
Figura 11 Bloque dentro de la tabla de atributos sprites



te ayudarán a comprender el tema. Como habrás notado, trabajar directamente sobre la memoria de video no es complicado, y puede ayudarnos a agregar efectos especiales a nuestros programas como mostramos en los listados de esta nota.

Aquí dejamos todo lo referido a la VRAM, nos encontraremos en próximos números, cuando hablemos de la VRAM para los modos 2 y 3.

Figura 12



```
10 SCREEN 1
20 CO=BASE(9):REM Base (9) da el
 comienzo de ;la tabla de patrone
s sprites para el modo 1.
30 FOR F=CO TO CO+7:REM Abarca l
os 8 primeros bytes de la tabla
40 VPOKE F,195:REM Definimos el
sprite para el plano 0 por haber
elegido los primeros 8 bytes de l
a tabla
50 NEXT
60 PUT SPRITE 0,(110,100),6,0:RE
M Colocamos el sprite en pantalla
70 FOR F=1 TO 400:NEXT:REM Retar
do para visualizar el sprite ante
s de cambiarlo
80 ATRI=BASE(8):REM La variable
'ATRI' tomara el valor del comien
zo de la tabla de atributos
90 VPOKE ATRI,10:REM Cambia el v
alor de la coordenada X
100 VPOKE ATRI+1,30:REM Cambia l
a coordenada Y
110 VPOKE ATRI+3,6:REM Cambia el
 color del sprite
```

Sortilegios

CARACTERES INVERTIDOS

El siguiente programa en código de máquina fue escrito por Daniel Galimberti, y nos brinda la posibilidad de trabajar con caracteres invertidos en screen 0. A continuación les damos el listado en basic necesario para cargar el programa en la memoria de la MSX. Los caracteres invertidos diseñados con este programa se borran apenas ejecutemos alguna orden SCREEN. Para recuperarlos, basta con hacer un DEF USR = D000 y un PRINT USR (0). (Por supuesto, apretando la tecla Return luego de cada una de estas instrucciones).

VELOCIDAD

Si dibujamos un rectángulo y luego queremos pintarlo con alguno de los 16 colores que nos proporciona nuestra MSX, normalmente nos inclinamos por utilizar las sentencias LINE y PAINT.

Figura 1

```
10 SCREEN 2
20 LINE(0,0)-(100,100),11,BF
30 GOTO 30
```

Figura 2

```
10 SCREEN 2
20 LINE(0,0)-(100,100),,B
30 PAINT (50,50),11
40 GOTO 40
```

Pero esta no es la mejor forma, pues consume demasiado tiempo.

Debemos tener presente que a la sentencia LINE se le puede agregar el parámetro BF que justamente, le indica al sistema que se trata de un rectángulo pintado en su interior.

En la figura 2, mostramos cómo sería el programa con LINE y PAINT, en cambio en la figura 1, solo se emplea LINE.

Al ejecutar ambos programas, notaremos la diferencia de ejecución entre ambos.

Esto es aconsejable tenerlo presente cuando realicemos programas con gráficos.

```
10  DATA3E,20,32,90,D0,3E,E,32,9
1,D0,21,8,A,11,58,D0,6,1A,C5,D5
20 DATACD,3E,D0,D1,C1,F5,1A,32,9
0,D0,13,1A,32,91,D0,13,F1,10,EB,2
1
30 DATA0,9,CD,3E,D0,3E,98,32,90,
D0,3E,F,32,91,D0,21,68,9,CD,3E
40 DATAD0,C9,6,8,ED,5B,90,D0,CD,
4A,0,2F,E5,CD,51,D0,E1,23,10,F4
50 DATAC9,D5,E1,CD,4D,0,13,C9,88
,8,E0,D,38,E,68,E,A0,8,A8,8
60 DATA98,8,E0,E,30,E,E8,E,40,E,
58,8,D8,8,10,E,D8,E,60,E
70 DATAC0,8,90,E,90,8,0,E,D0,8,7
8,E,E0,8,C8,8,78,8,90,F
80 FORA=&HD000TO&HD08B
90 READ B$
100 POKE A,VAL("&H"+B$)
110 NEXT
120 DEF USR=&HD000
130 PRINTUSR(0)
```

TEST MATEMATICO

Este programa está pensado para ayudar a repasar los cálculos matemáticos a los chicos que estén cursando la escuela primaria. Por medio de un menú que aparece en pantalla, se pueden seleccionar alternativas: repasar las sumas, restas, multiplicaciones, o bien practicarlas

Para quienes tienen una impresora conectada a su MSX, el software también les permite imprimir sobre papel las tablas de multiplicación

VARIABLES IMPORTANTES:

Z: opción escogida del menú

S%: número de respuestas correctas

N: número de cuentas en total

A: respuesta de la cuenta dada por el usuario

CLASE: EDUCATIVO

```
Menú
----
1- Multiplicación
2- Suma
3- Resta
4- Tablas de
   multiplicación
? 4
```

ESTRUCTURA DEL PROGRAMA:

10-250: pantalla de presentación
260-360: mensaje inicial
370-410: menú
420-430: elección de la opción
440-600: rutina de multiplicación
610-760: rutina de suma
770-930: rutina de resta
940-1300: rutina para las tablas de multiplicación

```
10 SCREEN 2,2,0:COLOR 4,15,1
20 KEY OFF
30 CLS
40 FOR T=1 TO 32:READ D
50 S$=S$+CHR$(D)
60 NEXT T
70 SPRITE$(1)=S$
80 PUT SPRITE 1,(0,192),7,1
90 DATA 7,13,30,13,5,1,15,17,17,
17,57,1,7,7,0,0
100 DATA 224,176,120,176,160,128
,240,72,72,72,92,64,112,112,0,0
110 A$="TEST MATEMATICO"
120 OPEN"GRP:" FOR OUTPUT AS#1
130 FOR VE=1 TO 7:PLAY"s8m900cde
fg":NEXT VE
140 FOR X=1 TO 255
150 PUT SPRITE 1,(X,5),1,1
160 Y=0:IF X/15=INT(X/15) THEN Y
=X/15
170 IF Y>0 THEN B$=MID$(A$,Y,1):
PRESET(X,5):PRINT#1,B$
180 NEXT X
190 PLAY"s8m900cdefg"
200 A$="   LOAD MSX"
210 FOR X=1 TO 255
220 PUT SPRITE 1,(X,30),1,1
230 Y=0:IF X/10=INT(X/10) THEN Y
=X/10
240 IF Y>0 THEN B$=MID$(A$,Y,1):
PRESET(X,30):PRINT#1,B$
250 NEXT X
260 SCREEN 0
270 LOCATE 9,3:PRINT"TEST MATEMA
TICO":LOCATE 9,4:PRINT"__________
________"
280 LOCATE 10,6:PRINT"LOAD MSX"
290 LOCATE 0,9
300 PRINT"El objetivo de este ju
ego es princi-"
310 PRINT:PRINT"palmente educati
vo. Enseña cuentas",,,"matemática
s simples."
320 PRINT:PRINT"Tú debes tratar
de contestarlas"
330 PRINT:PRINT"y obtendras el p
untaje merecido."
340 PRINT:PRINT"El puntaje se da
rá luego de 10 "
350 PRINT:PRINT"cuentas."
360 FOR X=0 TO 4500:NEXT
370 PLAY"s8m900cdefg"
380 CLS:COLOR 2
390 LOCATE 2,4:PRINT"Menú":LOCAT
E 2,5:PRINT"____"
400 PRINT"1- Multiplicación":PRI
NT:PRINT"2- Suma":PRINT:PRINT"3-
Resta":PRINT:PRINT"4- Tablas de m
ultiplicación"
410 PRINT:PRINT:INPUT Z
420 ON Z GOTO 440,610,770,940
430 IF Z<>1 OR Z<>2 OR Z<>3 OR Z
<>4 THEN 400
440 S%=0:N=0
450 COLOR 10
460 FOR Z=1 TO 10
470 CLS
480 X=INT(RND(1)*12+1)
490 Y=INT(RND(1)*12+1)
500 LOCATE 2,0:PRINTZ;"¿Cuánto e
s ";X;" x ";Y;"?"
510 PRINT:INPUT A
520 IF A=X*Y THEN LOCATE 2,4:PRI
NT"Muy bien!!":S%=S%+1:N=N+1
530 IF A<>X*Y THEN LOCATE 2,4:PR
INT"mala suerte!!":N=N+1:LOCATE 2
,8:PRINT"La respuesta correcta es
: ";X*Y
540 FOR A=0 TO 1000:NEXT
550 NEXT Z
560 CLS
570 LOCATE 2,4:PRINT"Tu puntaje
total sobre 10 fue: ";S%
580 FOR D=0 TO 2000:NEXT
590 PLAY"s8m900cdefg"
600 FOR Y=0 TO 1000:NEXT:RUN
610 S%=0:N=0
620 FOR Z=1 TO 10
630 CLS
640 X=INT(RND(1)*50+1)
650 Y=INT(RND(1)*50+1)
660 LOCATE 2,0:PRINT Z;" ¿Cuánto
 es ";X;" + ";Y;"?"
670 PRINT:INPUT A
680 IF A=X+Y THEN LOCATE 2,4:PRI
NT" Correcto!!":S%=S%+1:N=N+1
690 IF A<>X+Y THEN LOCATE 2,4:PR
INT" Mall !!":N=N+1:LOCATE 2,8:PR
INT" La respuesta era: ";X+Y
700 FOR A=0 TO 1000:NEXT
710 NEXT Z
720 CLS:COLOR 8
730 LOCATE 2,4:PRINT"Tu puntaje
total es; ";S%
740 FOR D=0 TO 2000:NEXT
750 PLAY"s8m900cdefg"
760 RUN
770 S%=0:N=0
780 FOR Z=1 TO 10
790 CLS:COLOR 7
800 X=INT(RND(1)*100+1)
810 Y=INT(RND(1)*100+1)
820 IF Y>X THEN 800
830 LOCATE 2,0:PRINTZ;"¿Cuánto e
s: ";X;" - ";Y;"?"
840 PRINT:INPUT A
850 IF A=X-Y THEN LOCATE 2,4:PRI
NT"Muy bien, te felicito":S%=S%+1
:N=N+1
860 IF A<>X-Y THEN LOCATE 2,4:PR
INT"Mal, mal!!":N=N+1:LOCATE 2,8:
PRINT"la respuesta era:";X-Y
870 FOR A=0 TO 1000:NEXT
880 NEXT Z
890 CLS:COLOR 6
900 LOCATE 0,4:PRINT"El puntaje
total que mereces es:";S%:PRINT"s
obre 10"
910 FOR D=0 TO 2000:NEXT
920 PLAY"s8m900cdefg"
930 RUN
940 CLS:COLOR 1
950 LOCATE 2,3:PRINT"¿Qué tabla
de multiplicación deseas repasar?
 (del 1 al 12)"
960 LOCATE 3,6:PRINT"Entra solo
el número":PRINT"
970 INPUT Z
980 CLS
990 FOR X=1 TO 12
1000 COLOR 8
1010 LOCATE 0,X:PRINT X:LOCATE 3
,X:PRINT"X":LOCATE 4,X:PRINTZ:LOC
ATE 7,X:PRINT"=":LOCATE 8,X:PRINT
 X*Z
1020 NEXT X
1030 LOCATE 0,14:PRINT"¿Querés o
btener un listado en papel de la
tabla?":LOCATE 0,16:PRINT"1-SI","
0-NO"
1040 INPUT P
1050 IF P=0 THEN PLAY"s8m900cdef
g":RUN
1060 IF P=1 THEN 1070
1070 IF Z=1 THEN A$="uno"
1080 IF Z=2 THEN A$="dos"
1090 IF Z=3 THEN A$="tres"
1100 IF Z=4 THEN A$="cuatro"
1110 IF Z=5 THEN A$="cinco"
1120 IF Z=6 THEN A$="seis"
1130 IF Z=7 THEN A$="siete"
1140 IF Z=8 THEN A$="ocho"
1150 IF Z=9 THEN A$="nueve"
1160 IF Z=10 THEN A$="diez"
1170 IF Z=11 THEN A$="once"
1180 IF Z=12 THEN A$="doce"
1190 LPRINT CHR$(&H1B)+"c":LPRIN
TCHR$(&H12)
1200 LPRINT "Tabla del ";A$
1210 FOR X=1 TO 9
1220 LPRINT CHR$(&H1B)+"c"
1230 LPRINT " ";X;" x ";Z;" = ";
X*Z
1240 NEXT X
1250 FOR X=10 TO 12
1260 LPRINT CHR$(&H1B)+"c"
1270 LPRINT X;" x ";Z;" = ";X*Z
1280 NEXT X
1290 PLAY"s8m900cdefg"
1300 RUN
```

Clase: comercial

FICHERO MSX

Este es un programa Basic de alta calidad que nos permite crear un fichero para lo que necesitemos. Así, gracias a nuestra MSX, podemos tener clasificada y ordenada nuestra discoteca o biblioteca, o una lista de proveedores, o nuestro índice telefónico, así como las direcciones de nuestros amigos y conocidos. Seguro que una vez que empiecen a usar el fichero, le encontrarán mil utilidades más.

Los datos pueden ser almacenados en diskette si es que la máquina tiene una disquetera conectada. De lo contrario, deberán ser almacenados en cassettes. Si se usa disquetera, al correr el programa se deben ignorar los mensajes en relación al cassette que aparecerán en la pantalla (poner Play, etcétera), y cuando aparezcan sólo habrá que preparar el diskette y presionar cualquier tecla. El programa acepta hasta 300 fichas, los

```
MENU DEL PROGRAMA
*****************
1. CREACION DE UN FICHERO
2. MODIFICACION DEL FICHERO
3. CREAR NUEVAS FICHAS
4. INTERROGAR
5. LISTADO ALFABETICO
6. LISTADO DE TODAS LAS FICHAS
7. FINAL
¿Cuál es su opción? █
```

tipos de datos que irán en cada ficha. Podemos decidir al usar el programa, cuando la máquina nos pregunte por los ítems o títulos. Así, si queremos hacer un índice telefónico, llamaremos a los títulos Nombre, Apellido, Dirección, etcétera. Puede haber hasta 15 títulos por ficha, pero los títulos son los mismos para todas las fichas.

El programa tiene un menú desde donde se puede ir a las partes destinadas a creación de las fichas, modificaciones, ordenamiento, visualización, impresión, etcétera.

Recomendamos al terminar de trabajar con un fichero, antes de empezar a trabajar con otro, ir primero a la opción 7 del menú (Fin) y luego volver a poner Run.

Las variables más importantes son:

NF$ ... Nombre del fichero
NF ..... Cantidad de fichas
LA$(N) Títulos de los ítems
ND .... Cantidad de ítems
N$(M,N) Datos

Damos a continuación los números de línea en que comienzan las principales rutinas del programa.

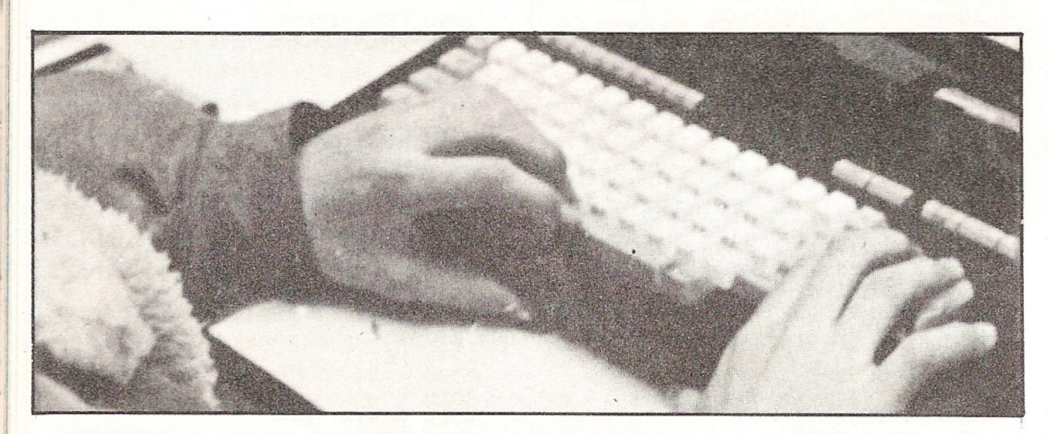
60 - creación de un nuevo fichero
1420 - display menú principal
1660 - modificación de las fichas ya existentes
1890 - creación de nuevas fichas
2080 - averiguar y buscar datos en las fichas ya existentes
2680 - ordenamiento alfabético
3260 - listado del contenido de las fichas
2460 - finalización

```
10 KEYOFF:SCREENO:WIDTH37
20 CLEAR 1000
30 MAXFILES=5
40 TIME=0
50 COLOR 1,9
60 CLS:GOTO 1420
70 CLS
80 PRINT"******************"
90 PRINT"*                *"
100 PRINT"*'  FICHERO     *"
110 PRINT"*               *"
120 PRINT"*  POR GUSTAVO  *"
130 PRINT"*  FAIGENBAUM   *"
140 PRINT"******************"
150 RETURN
160 'CREACION DE LAS VARIABLES
170 GOSUB70
180 INPUT "NOMBRE DEL FICHERO A
CREAR";NF$:IFLEN(NF$)=0THEN180
190 INPUT "NUMERO DE ITEMS (máxi
mo 15)";ND
200 IFND>15THEN190
210 IFND<=0THEN190
220 INPUT "NUMERO DE FICHAS (Máx
imo 300)";NF
230 IFNF>300ORNF<1THEN220
240 PRINT"Los datos entrados son
 correctos      (s/n)"
250 R$=INPUT$(1)
260 IFR$="n"ORR$="N"THENPRINT"re
comience":GOTO 160
270 IFR$><"s"ANDR$><"S"THEN250EL
SE330
280 'creacion de los vectores
290 IF RW$="*"THEN320
300 DIM LA$(ND),N$(NF,ND),XG$(NF
+1),XH(NF+1)
310 RW$="*"
320 RETURN
330 'creación de los títulos
340 GOSUB 70:GOSUB 280
350 PRINT"Máximo por título: 15
caracteres"
360 FOR I=1TOND
370 PRINT"Título";I;:INPUT LB$
380 IFLB$=""THEN370
390 IFLEN(LB$)>15THEN370
400 FORI1=1TO15-LEN(LB$)
410 LB$=LB$+"."
420 NEXTI1
430 LA$(I)=LB$:LB$=""
440 NEXTI
450 'visualizacion de los titulo
s
460 GOSUB 70
470 FORI=1TOND
480 PRINTI;" ";LA$(I)
490 NEXTI
500 PRINT"Los datos entrados son
 correctos      (s/n)"
510 R$=INPUT$(1)
520 IFR$="n"ORR$="N"THENPRINT"re
comience":GOTO 540
530 IFR$><"s"ANDR$><"S"THEN510EL
SE630
540 PRINT"Qué título quiere camb
iar (número)":INPUT R
550 IFR>NDORR<1THEN 540
560 PRINT"Titulo";R;:INPUT LB$
570 IF LEN(LB$)>15ORLEN(LB$)=0TH
EN560
580 FOR I1=1TO 15-LEN(LB$)
590 LB$=LB$+"."
600 NEXT I1
610 LA$(R)=LB$:LB$=""
620 GOTO470
630 'creacion de las fichas
640 DF=1
650 FORI=DFTO NF
660 GOSUB 70
670 PRINT"No te olvides de tecle
ar fin dentro  de ";LA$(1);" si t
erminaste"
680 PRINT"FICHA NUMERO";I
690 FORB=1TOND
700 PRINTB;:PRINTSPC(3-LEN(STR$(
B)))LA$(B);
710 INPUTN1$
720 IFN1$=""THEN 700
730 IF N1$="FIN"ORN1$="fin"THEN9
20
740 N$(I,B)=N1$:N1$=""
750 NEXTB
760 GOSUB70
770 FORB=1TOND
780 PRINTB;:PRINTSPC(3-LEN(STR$(
B)))LA$(B);
790 PRINTN$(I,B)
800 NEXTB
810 PRINT"¿datos correctos? (s/n
)"
820 R$=INPUT$(1)
830 IFR$="S"ORR$="s"THEN900
840 IFR$><"n"ANDR$><"N"THEN820
850 PRINT"Qué número":INPUT B2
860 IFB2<1ORB2>NDTHEN850
870 PRINTB2;SPC(3-LEN(STR$(B)))L
A$(B2);
880 INPUT N$(I,B2)
890 GOTO 760
900 NU=I
910 NEXTI
920 'guardar en cassette o diske
tte
930 CLS:GOSUB 1100
940 IFNU=0THENNU=NF
950 OPEN NF$ FOR OUTPUT AS 1
960 NU$=STR$(NU):ND$=STR$(ND)
970 PRINT#1,NU$:PRINT#1,ND$
980 FORI=1TOND
990 PRINT#1,LA$(I)
1000 NEXT I
1010 FORI=1TONU
1020 N$(I,0)=STR$(I)
1030 FORB=0TOND
1040 PRINT#1,N$(I,B)
1050 NEXT B
1060 NEXT I
1070 CLOSE 1
1080 GOTO 3390
1090 GOTO 1420
1100 'introducción del cassette
para registro
1110 PRINT"REBOBINE EL CASSETTE
Y PRESIONE UNA  TECLA"
1120 R$=INPUT$(1):MOTOR ON
1130 PRINT"PONGA AL CASSETTE EN
'PLAY' Y 'REC' Y PRESIONE UNA TEC
LA"
1140 R$=INPUT$(1)
1150 RETURN
1160 'LECTURA DEL CASSETTE O DIS
KETTE
1170 IFNF$=""THENINPUT"NOMBRE DE
L FICHERO";NF$
```

```
1180 GOSUB 1350
1190 OPEN NF$ FOR INPUT AS 1
1200 INPUT #1,NU$:INPUT#1,ND$
1210 NF=VAL(NU$)
1220 ND=VAL(ND$)
1230 IFRV$="**"THEN GOSUB 1940
1240 GOSUB 280
1250 FORI=1TOVAL(ND$)
1260 INPUT#1,LA$(I)
1270 NEXTI
1280 FORI=1TOVAL(NU$)
1290 FORB=0TOVAL(ND$)
1300 INPUT#1,N$(I,B)
1310 NEXTB
1320 NEXTI
1330 CLOSE1:RETURN
1340 'introduccion del cassette
1350 GOSUB70
1360 PRINT"Rebobine el cassette.
 Cuando esté    listo presione un
a tecla"
1370 R$=INPUT$(1)
1380 MOTOR ON
1390 PRINT"Ponga al cassette en
'PLAY' y luego  presione una tecl
a"
1400 R$=INPUT$(1)
1410 RETURN
1420 CLS:'presentación
1430 PRINT
1440 PRINT"      MENU DEL PROGRAM
A"
1450 PRINT"      ****************
*"
1460 PRINT
1470 PRINT"1. CREACION DE UN FIC
HERO"
1480 PRINT
1490 PRINT"2. MODIFICACION DEL F
ICHERO"
1500 PRINT
1510 PRINT"3. CREAR NUEVAS FICHA
S"
1520 PRINT
1530 PRINT"4. INTERROGAR"
1540 PRINT
1550 PRINT"5. LISTADO ALFABETICO
"
1560 PRINT
1570 PRINT"6. LISTADO DE TODAS L
AS FICHAS"
1580 PRINT
1590 PRINT"7. FINAL"
1600 PRINT
1610 PRINT"¿Cuál es su opción";
1620 INPUTR$
1630 IFR$>"7"ORR$<"1"THEN1420
1640 CLS:ON VAL(R$)GOTO 160,1660
,1890,2080,2680,3260,3460
1650 'modificación del fichero
1660 IFND>1THEN1680
1670 GOSUB 1160
1680 GOSUB 70
1690 PRINT
1700 PRINT"si terminaste pedí ot
ra ficha y       tecleá un 99
                    ¿Qué ficha qu
erés modificar";
1710 INPUT MF$
1720 MF=VAL(MF$)
1730 IF MF<1ORMF>NFTHEN1700
1740 GOSUB 70
1750 FORI=1TOND
1760 PRINTSPC(3)I;LA$(I);N$(MF,I
)
1770 NEXTI
1780 PRINT"si terminaste las mod
ificaciones     entrá un 99"
1790 PRINT"Qué item vas a modifi
car"
1800 INPUT B2$
1810 B2=VAL(B2$)
1820 IFB2=99THEN920
1830 IFB2<1ORB2>NDTHEN1780
1840 PRINTSPC(3);LA$(B2);:INPUT
N$(MF,B2)
1850 PRINT"¿datos correctos? (S/
N)"
1860 R$=INPUT$(1)
1870 IFR$="n"ORR$="N"THEN1780
1880 IFR$><"s"ANDR$><"S"THEN1860
ELSE1700
1890 'crear nuevas fichas
1900 RUN 1910
1910 RV$="**"
1920 CLS:GOSUB 1160
1930 GOTO650
1940 'numero de fichas a crear
1950 GOSUB 70
1960 PRINT"hay ";NF;"fichas"
1970 PRINT"cuantas querés crear"
1980 INPUT NA
1990 IF NA=0THENPRINT"fin del tr
abajo":GOTO 1420
2000 IF NA+NF>300THENPRINT"impos
ible":GOTO 1970
2010 PRINT"seguro? (s/n)"
2020 R$=INPUT$(1)
2030 IFR$="S"ORR$="s"THEN2050
2040 IF R$="n"ORR$="N"THEN1950EL
SE2020
2050 NF=NF+NA:DF=NF-NA+1
2060 RETURN
2070 'interrogar las fichas
2080 IF NF>1THEN2100
2090 GOSUB 1160
2100 GOSUB 2300
2110 IFCH$="2"THEN2390
2120 GOSUB70
2130 'interrogación por número d
e ficha
2140 INPUT"Qué ficha querés list
ar";NL
2150 IF NL<1ORNL>NFTHENGOTO 2140
2160 GOSUB 3150
2170 OPEN TE$FOR OUTPUT AS2
2180 PRINT#2,"FICHA NUMERO";NL
2190 FORB=1TO ND
2200 PRINT#2,SPC(3)LA$(B);N$(NL,
B)
2210 NEXTB
2220 CLOSE2
2230 PRINT"presione una tecla pa
ra continuar"
2240 R$=INPUT$(1)
2250 PRINT"Otra ficha? (s/n)"
2260 R$=INPUT$(1)
2270 IFR$="S"ORR$="s"THEN2140
2280 IFR$="n"ORR$="N"THEN1420
2290 GOTO 2250
2300 'eleccion de la interrogaci
ón
2310 GOSUB70
2320 PRINT" 1. número de ficha"
2330 PRINT" 2. un dato"
2340 PRINT"OPRIMA EL NUMERO DE S
U ELECCION (1/2)"
2350 INPUT R$
2360 IFR$="1"THEN CH$="1":RETURN
2370 IFR$="2"THENCH$="2":RETURN
2380 GOTO 2350
2390 'INTERROGACION POR DATOS
2400 GOSUB 70
2410 FORI=1TOND
2420 PRINTSPC(3)I;LA$(I)
2430 NEXTI
2440 PRINT"Que criterio de selec
ción deseás"
2450 INPUT W
2460 IF W<1ORW>NDTHEN2440
2470 PRINT"¿Qué dato buscás?"
2480 INPUT DR$
2490 IFDR$=""THEN2470
2500 FORI=1TONF
2510 IF N$(I,W)=DR$THENGOSUB2570
:NEXTI:GOTO2540
2520 NEXTI
2530 PRINT"No hay más datos ";DR
$:PRINT"en la memoria bajo el tít
ulo";W
2540 PRINT"presione una tecla"
2550 A$=INPUT$(1)
2560 GOTO 1420
2570 'selección
2580 GOSUB 3150
2590 OPEN TE$FOR OUTPUT AS 4
2600 PRINT#4,"FICHA NUMERO ";N$(
I,0)
2610 FORB=1TOND
2620 PRINT#4,SPC(3)LA$(B);N$(I,B
)
2630 NEXTB
2640 CLOSE4
2650 PRINT"pulse cualquier tecla
 para seguir"
2660 R$=INPUT$(1)
2670 RETURN
2680 'listado alfabetico
2690 IF ND>1THEN2710
2700 GOSUB1160
2710 GOSUB 70
2720 PRINT:PRINT:PRINT
2730 FORI=1TOND
2740 PRINTSPC(3)I;LA$(I)
2750 NEXTI
2760 PRINT"Qué criterio de orden
amiento"
2770 INPUTW
2780 IFW>NDORW<1THEN2710
2790 GOSUB2940
2800 GOSUB3150
2810 OPEN TE$FOR OUTPUT AS3
2820 FORI=1TOXI
2830 GOSUB70:PRINT
2840 PRINT#3,"orden";I
2850 PRINT#3,"número",N$(XH(I),0
)
2860 FORB=1TOND
2870 PRINT#3,SPC(3)LA$(B);N$(XH(
I),B)
2880 NEXTB
2890 PRINT"para seguir presione
una tecla"
2900 R$=INPUT$(1)
2910 NEXTI
2920 CLOSE#3
2930 GOTO1420
2940 'alfabetico
2950 XD=NF
2960 XI=0
2970 FORI=1TONF
2980 XI=XI+1:XG$(XI)=N$(XI,W):XH
(XI)=I
2990 NEXTI
3000 K=XI
3010 XD=0
3020 FORI=1TOK-1
3030 IFXG$(I+1)<XG$(I)THEN3050
3040 GOTO 3120
3050 XJ$=XG$(I)
3060 XG$(I)=XG$(I+1)
3070 XG$(I+1)=XJ$
3080 XF=XH(I)
3090 XH(I)=XH(I+1)
3100 XH(I+1)=XF
3110 XD=1
3120 NEXTI
3130 IFXD=1THENK=K-1:GOTO3010
3140 RETURN
3150 'seleccion para lectura
3160 'pantalla e impresora
3170 GOSUB70
3180 PRINT"Desea listar:"
3190 PRINT" 1. en pantalla"
3200 PRINT" 2. en impresora"
3210 INPUT R$
3220 IFR$="1"THENTE$="crt:":RETU
RN
3230 IFR$="2"THENTE$="lpt:":RETU
RN
3240 GOTO 3180
3250 'listado de fichas
3260 IFND>1THEN3280
3270 GOSUB1160
3280 GOSUB3190
3290 OPEN TE$FOR OUTPUT AS 5
3300 FORD=1TONF:GOSUB70:PRINT#5,
"ficha número";D
3310 FORB=1TOND
3320 PRINT#5,SPC(3)LA$(B);N$(D,B
)
3330 NEXTB
3340 PRINT"Para continuar presio
ne una tecla"
3350 R$=INPUT$(1)
3360 NEXTD
3370 CLOSE5
3380 GOTO 1420
3390 'copia del fichero
3400 GOSUB70
3410 PRINT"Quiere una copia del
fichero";
3420 R$=INPUT$(1)
3430 IFR$="s"ORR$="S"THEN920
3440 IFR$="n"ORR$="N"THEN1420
3450 GOTO 3420
3460 GOSUB70
3470 PRINT"chau"
3480 END
```

UNA VENTANA EN LA PANTALLA

Gracias a la instrucción LD, una de las más simples del assembler, podremos hacer un programa que abre una ventana.



LD es la abreviatura de la palabra LoaD, que en inglés significa 'cargar'. La forma más fácil de entender la función de esta instrucción es compararla con la instrucción Basic LEFT, que permite poner un valor en una variable.

LET A=10
o bien:
A=10
En Assembler, el equivalente seria:
LD A,10

Claro que muchas personas estarán pensando cómo hacer para entrar esta última instrucción en sus máquinas, que sólo aceptan órdenes Basic y que darían un Syntax Error ante cualquier LD. Les pedimos un poco de paciencia: ya veremos cómo entrar estas órdenes a nuestra MSX sin necesidad de ningún programa ensamblador ni nada similar (aunque si lo tenemos mejor).

Terminando de definir a la instrucción LD, digamos que permite colocar números en los registros internos del microprocesador (registro: equivalente a la variable del Basic), copiar el dato de un registro en otro o en alguna dirección de memoria y copiar el contenido de una dirección de memoria en un registro. Esto quiere decir que si bien en Assembler trabajamos con sólo unos pocos registros ya creados (generalmente 7 u 8, a diferencia del Basic donde podemos 'inventar' nuestras propias variables) gracias a la instrucción LD podemos cargar en ellos 'trocitos' de la memoria de nuestra máquina (y viceversa).

Estos 'trocitos' a los que nos referimos son los bytes. Imaginémoslos como pequeñas cajitas, cada una de las cuales tiene un nombre y en cuyo interior se puede guardar un número entre 0 y 255. El nombre de estos bytes que forman la memoria también es un número que puede estar entre 0 y 65535.

Pero veamos ahora las instrucciones LD que tenemos para cargar un número en un registro. (ver figura 1).

Figura 1

ISNTRUCCION	CODIGO
LD A,dato	3E
LD B,dato	06
LD C,dato	0E
LD D,dato	16
LD E,dato	1E
LD H,dato	26
LD L,dato	2E

En realidad, lo que entramos en la máquina no es la instrucción LD directamente sino el código que la representa. El código no es más que un número, pero en vez de estar representado en el sistema decimal (dígitos del 0 al 9) está en hexadecimal (dígitos del 0 a la F). No se asusten quienes no conozcan este sistema de numeración, la computadora lo entiende perfectamente con tal de poner un &H antes de un número hexadecimal.

Vamos a hacer ahora un programa en assembler que cargue en el registro B el número 0 y en el C el 128. Todo programa en assembler debe ser ubicado en la memoria de la máquina. El nuestro lo pondremos a partir del byte número 50000.

PROGRAMA:

```
LD  B,0
LD  C,128
RET
```

La instrucción RET (abreviatura de RETurn) significa que nuestro programa ha terminado (equivalente en Basic a END o STOP). En assembler, un programa al ser ejecutado NO para automáticamente después de la última instrucción, sino que si no hay un RET seguirá intentando interpretar las siguientes regiones de memoria, que no tienen sentido. En este caso, el programa no se detiene ni siquiera con las teclas Control y Stop, y habría que apagar y volver a encender la computadora.

Vamos a transformar a las instrucciones en sus códigos.

Como verán, las instrucciones LD REGISTRO, DATO ocupan 2 bytes, uno para la instrucción y otro para el dato (que a la derecha lo hemos pasado a hexadecimal, = 128=&H80). Podemos colocar ahora nuestro programa a partir de la memoria 50000 por medio de pokes:

```
POKE 50000,&H06
POKE 50001,&H00
POKE 50002,&H0E
POKE 50003,&H80
POKE 50004,&HC9
```

En realidad no tiene mucha gracia ejecutar este programa, porque tenemos que hacerlo desde Basic, y no sabemos ningún método para comprobar si anduvo bien, es decir, si los registros se cargaron con los valores que pusimos. De todos modos, si alguien tiene ganas de hacerlo, el equivalente al RUN se logra tecleando:

DEF USR = 50000
(que equivale a decirle a la máquina: nuestro programa empieza en la posición de memoria número 50000)

A = USR (0)
(que es como poner RUN al programa que está en la posición de memoria que pusimos en **DEF USR**).
Si todo va bien, el programa se ejecutará rapidisimo, y el Ok y el cursor aparecerán en la pantalla. Esto significa que el programa que hicimos en assembler se ha ejecutado como si fuera una subrutina del Basic, volviendo, al finalizar, a donde estaba antes de empezar.
También podemos cargar directamente pares de registros. En este caso, cada instrucción tiene 3 bytes, 1 para el código de operación y 2 para el número que tendrá 16 bits.

INSTRUCCION	**CODIGO**
LD BC,dato	**01(+ 2 bytes para el dato)**
LD DE,dato	**11 (+ 2 bytes para el dato)**
LD HL,dato	**21 (+ 2 bytes para el dato)**

Hagamos el programa

Memoria	**Assembler**	**Código de Máquina**
MEM. 50000	**LD BC,&H002B**	**012B00**
MEM. 50003	**RET**	**C9**

Noten que cuando pasamos un número de 2 bytes (como el hexadecimal 002B que cargamos en el par BC), al escribirlo en código de máquina debemos 'quebrarlo' en dos partes (la más significativa y la menos significativa) colocando primero la menos significativa (2B) y luego la más significativa (00).
Veremos ahora las instrucciones que nos permiten leer y escribir posiciones de memoria, lo que nos permite tener a nuestra disposición toda la memoria RAM

En Basic se puede escribir:
X = PEEK (&HD000)
que significa que la variable X asumirá un valor entre 0 y 255 que corresponde al que esta en la posición de memoria (en hexadecimal)
&HD000. En assembler existe una instrucción análoga, que funciona solo con el registro A:
LD A,(D000)
cuyo código es 3A más dos bytes para la dirección. Los paréntesis indican esta idea: copiamos en el A el contenido de la memoria D000. La instrucción LD A,D000 (sin paréntesis) no es válida porque no podemos colocar el número D000 (de dos bytes) en un único registro (donde cabe como máximo un 255).
Esta instrucción equivale a 3 bytes: una para el código de operación (en este caso 3A) y dos para decir cual es la dirección cuyo contenido vamos a colocar en el registro A.

Inversamente, existe una instrucción parecida que nos permite escribir en una posición de memoria cualquiera el contenido del registro A. El código de esta instrucción es 32 y también lleva dos bytes para la dirección. Observen este programa.

ASSEMBLER	**CODIGO DE MAQUINA**
LD A, (C000)	**3A00C0**
LD (C001),A	**3200C1**
RET	**C9**

El programa carga primero en el registro A el contenido de la memoria C000 y luego lo copia en la memoria C001
Podemos probar cargar el programa a partir de la posición de memoria D000 por medio de pokes:

```
POKE &HD000, &H3A
POKE &HD001,0
POKE &HD002,&HC0
POKE &HD003,&H32
etc.
```

Podemos ejecutarlo y verificar si funcionó. Para ello, primero carguemos un número en la memoria &HC000, por ejemplo:
POKE &HC000,208
Luego ejecutemos nuestro programa
DEF USR = &HD000
A = USR(0)
Finalmente, verifiquemos si el programa funcionó, haciendo:
PRINT PEEK (&HC001)
Si todo fue bien, el resultado de esta instrucción debería ser 208.
Vamos a aplicar ahora los conocimientos que acabamos de adquirir para crear una 'ventanita' en nuestro televisor.
La posición de memoria F3B0 de nuestra máquina tiene un número que indica la cantidad de columnas con las que trabajamos en la pantalla de texto. La F3B1 guarda la cantidad de filas (o renglones) en que trabajamos. Si las modificamos (desde assembler) podremos aumentar o disminuir 'a piaccere' el tamaño de nuestra pantalla.
Probemos darle un ancho de 10 columnas y un alto de 12 filas.

ASSEMBLER	**CODIGO DE MAQUINA**
LD A,10	**3E0A**
LD (F3B0),A	**32B0F3**
LD A,12	**3E0C**
LD (F3B1),A	**32B1F3**
RET	**C9**

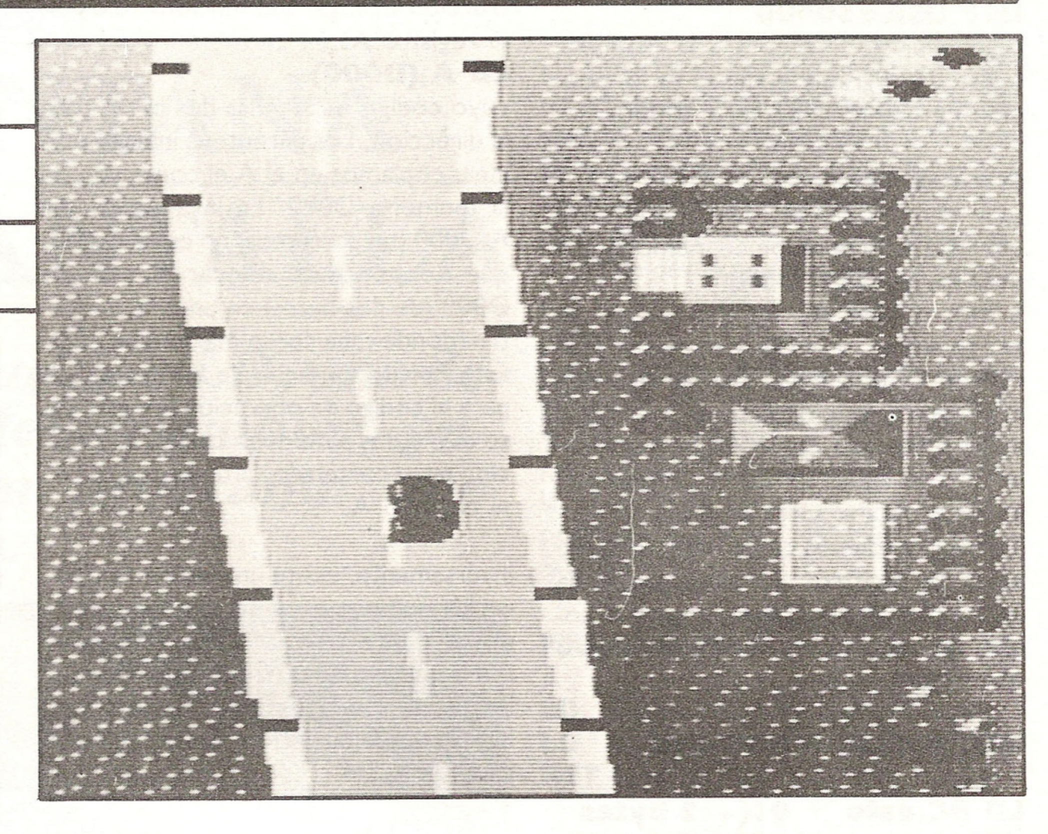
Para cargarlo:

```
POKE &HD000,&H3E
POKE &HD001,&H0A
POKE &HD002,&H32
POKE &HD003,&HB0
POKE &HD004,&HF3
POKE &HD005,&H3E
POKE &HD006,&H0C
POKE &HD007,&H32
POKE &HD008,&HBi
POKE &HD009,&HF3
POKE &HD00A,&HC9
```

Por fin hemos hecho un programa en assembler que tiene una utilidad práctica real y que nos permite ampliar las posibilidades de nuestro computador. Si querés seguir investigando, probá cambiar los valores para obtener otros tamaños de pantalla. Fijate que pasa con los números muy chicos o muy grandes, y estate preparado para tener que soportar algunas sorpresas desagradables (tener que apagar la máquina y perder un programa). Pero el riesgo está justificado: en Assembler, probar es la mejor forma de aprender.

GUSTAVO FAIGENBAUM

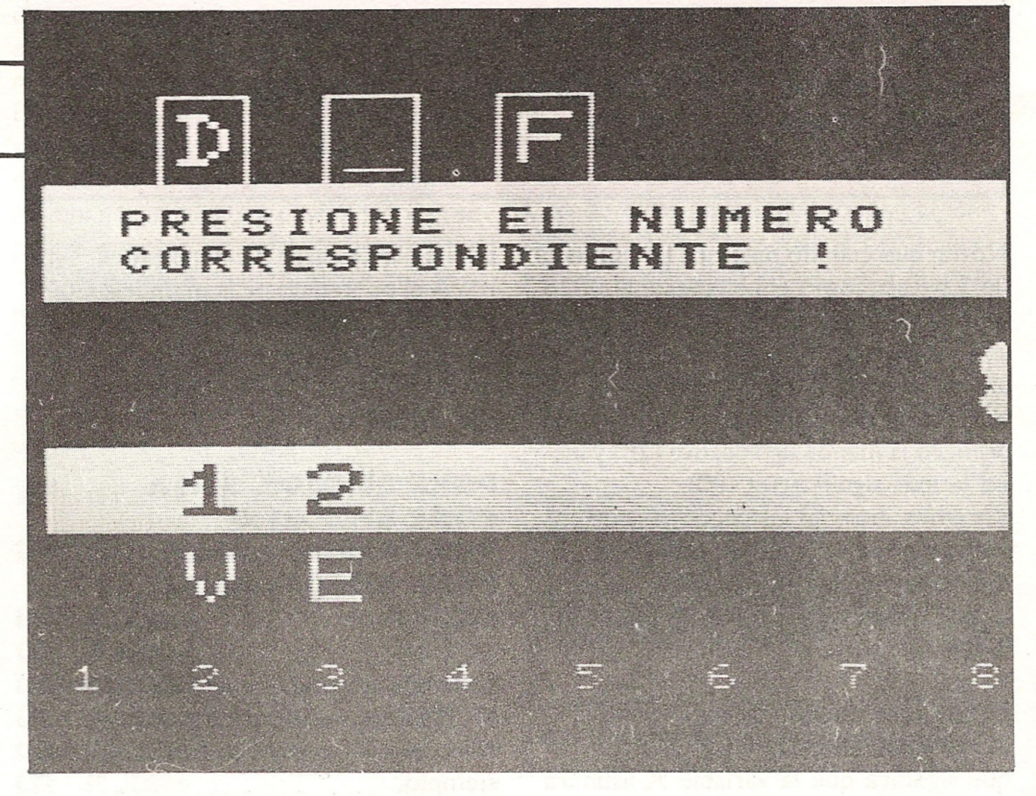
Road Fighter



CREATIVIDAD: 8
PRESENTACION: 8
ATRACCION: 9
GRAFICOS: 8
SONIDO: 8
TIPO: ENTRETENIMIENTO
PRODUCE: MICROBYTE

A cualquiera de nosotros nos gustaría participar en un gran Rally Internacional de automovilismo.
Partiremos de la última fila (para darles un poco de ventaja) y por medio de nuestra pericia conductiva, iremos superando a nuestros adversarios en la carrera, con los riesgos que implica una competencia de este tipo.
Podremos acelerar, hacer rebajes de velocidad y, de no hacerlos en forma correcta, nos despitaremos con las lógicas consecuencias.
Un factor importante será el combustible ya que nuestro mal manejo nos puede hacer gastarlo de más y ésto nos impedirá arribar a feliz término a la llegada.
El gran premio está dividido en distintas etapas con hermosos y cambiantes paisajes, lo que hace más apasionante el juego.
Si no se cumple una etapa hay que comenzar todo de nuevo.
Al llegar a la última parte y ganarla nos sentiremos con esa modestia que sólo los grandes pueden tener.

Números y letras



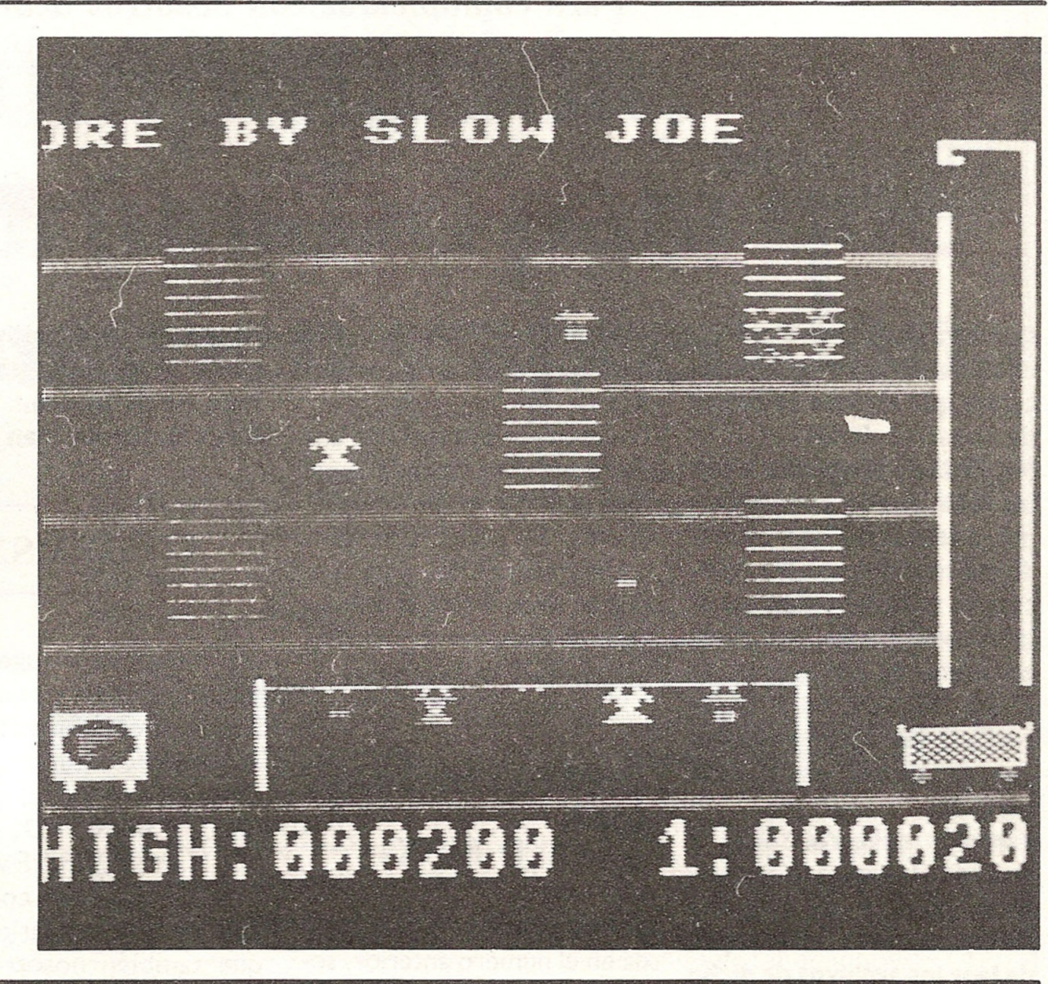
CREATIVIDAD: 7
PRESENTACION: 7
ATRACCION: 7
GRAFICOS: 8
SONIDO: 8
TIPO: EDUCATIVO
PRODUCE: MICROBYTE

Sentarse frente a la computadora para aprender el abecedario puede resultar una tarea interesante, si a esta propuesta se la combina con una ejercitación de la memoria, con el agregado de poder conocer sin ningún esfuerzo los números del 1 al 10 y la relación que existe entre los números y las letras.

Con una buena gráfica los más chicos de la familia podrán estudiar asistidos por la computadora lo que a veces lleva horas aprender en la escuela o hacer un repaso de las mismas.
Ellos con todo derecho verán a la máquina como a una infaltable amiga de estudios.
Al finalizar la clase se podrá ver cómo el chico avanza en sus estudios. También es interesante saber que en abecedario la letra ñ existe para las computadoras MSX.

Mr. Wong's Loopy Laundry

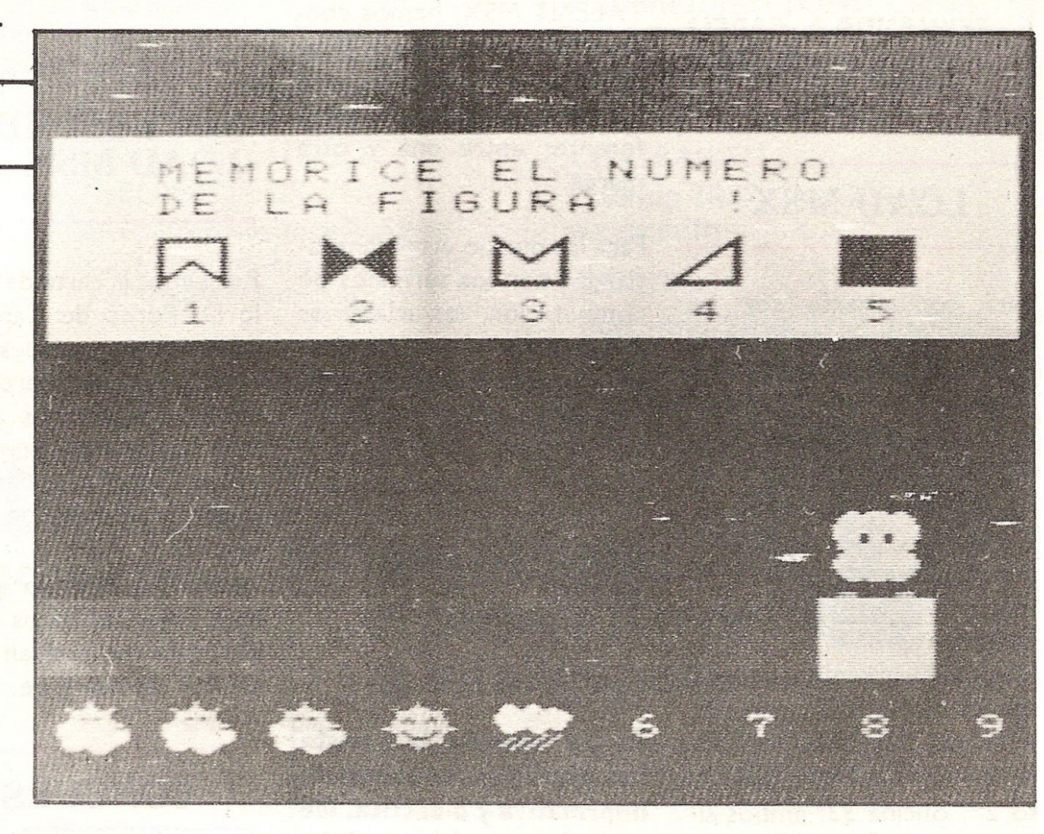
CREATIVIDAD: 8
PRESENTACION: 8
ATRACCION: 9
GRAFICOS: 8
SONIDO: 8
TIPO: ENTRETENIMIENTO
PRODUCE: MICROBYTE



Mr. Wong es un simpático chinito cuya misión es sortear un largo y peligroso camino en busca de la ropa de sus clientes, para poder realizar su trabajo en la lavandería.
A todo esto sus herramientas, como la plancha o el vapor que produce ésta, serán sus enemigos.
Recorrerá un peligroso camino de persecución en tan loable tarea y eso no será demasiado fácil.
Para los fanáticos de las persecusiones este soft es ideal. Con un buen diseño gráfico, brinda originalidad en los distintos escenarios donde Mr. Wong se sentirá casi inexorablemente atrapado por su vieja pero aún más traicionera plancha.

Figuras y números

CREATIVIDAD: 8
PRESENTACION: 8
ATRACCION: 9
GRAFICOS: 9
SONIDO: 9
TIPO: EDUCATIVO
PRODUCE: MICROBYTE



Este programa dedicado a los más pequeños computómanos, demuestra claramente que un software educativo no tiene por qué ser aburrido.
En forma de juego, nos obligará a recordar cada una de las figuras representadas y asociadas a un número determinado.

Con él, padres e hijos podrán practicar y pasar momentos divertidos.
En sí trata de agilizar nuestra capacidad de memoria visual, usando como herramientas, figuras muy similares y a la vez provoca que los más pequeños ejerciten o aprendan las formas y relaciones entre los números en forma extremadamente sencilla y práctica.
Excelentes colores, formas y sonido contribuyen a que esto sea, como dijimos al principio, un entrenimiento educativo.

Para comunicarse con nosotros deben escribirnos a "Load Revista para Usuarios de MSX", Paraná 720, 5to. Piso, (1017) Cap. Federal.

AGENCIAS DE NOTICIAS

Quisiera felicitarlos por su excelente revista y agradecerles que alguien se ocupe de nuestras dudas.
Me gustaría preguntar si con mi MSX puedo comunicarme, a través del correspondiente modem, con las agencias de noticias extranjeras tales como las que describen en números anteriores de LOAD MSX y K 64 y dónde realizar las correspondientes gestiones para este servicio.
También desearía saber si puedo colocar en la disquetera DPF-550 un disquete IBM y si la MSX sería capaz de leer los archivos de dicho disquete.

FERNANDO A. GARCIA
ROSARIO

LOAD MSX

Para comunicarte con las agencias de noticias, están las bases de datos **DELPHI**, servicio que brinda SISCOTEL, **THE SOURCE** y **COMPUSERVE,** servicio del INFOTEL.
Con ambas bases podrán comunicarse con agencias de noticias nacionales y extranjeras.
SISCOTEL atiende al público en Rivadavia 822, piso 1°, mientras INFOTEL lo hace en Bartolomé Mitre 912, piso 2°, oficina 33; ambos situados en la Capital Federal.
Pero si queremos encontrar más datos sobre cualquiera de las dos bases, podrán encontrarlos en los avisos que ambas publican en la revista K 64.
Los archivos escritos con una computadora profesional IBM pueden ser leídos o impresos en papel con otra computadora MSX. Esta ventaja se debe a que el formateo de disco entre una IBM y una MSX es la misma.

COMPATIBILIDAD MSX —LOAD MSX—

En relación a la carta publicada en el número anterior, sobre la compatibilidad entre una CANNON MSX y una TALENT MSX, hemos continuado investigando y llegamos a la conclusión de que la memoria RAM ocupa slots diferentes entre una y otra marca.

Para los que no están familiarizados con los términos de computación, les aclaramos que un "slot" es un área donde se puede conectar tanto memoria RAM como ROM.

COMENTARIO DE SOFT

En mi opinión la revista cumple con creces su misión informativa y didáctica. Me permito sugerirles incluir, aunque sea en forma muy sintética, la utilidad del software de gestión, utilitario y educativo, pues estoy informado de su disponibilidad, pero no de su uso.
Jorge H. Sardin - San Miguel

LOAD MSX

Agradecemos tu sugerencia, y como podrás notar, ya la hemos implementado, pues en números anteriores explicamos la utilización del LOGO, TASWORD, MSX-PLAN, MSX-WRITE y otros.
Esperamos poder continuar con estos comentarios, porque también nosotros creemos que son sumamente útiles al lector.

SIN RESPUESTA LOAD MSX

Recibimos la carta de Marcelo D. Porco de Cap. Fed., con dos inquietudes sobre unos programas. Pero no pudimos contestarlas porque falta aclarar a que número de revista corresponden a sus dudas y el nombre de los programas.
Pedimos a Marcelo que nos haga llegar los datos detallados, para que puedan ser resueltas rápidamente.

CONCURSO

Los felicito por vuestra revista, que considero de lo más original y completa. Soy un usuario de una MSX, puesto que un amigo la posee y además, porque estudio Analista de Sistemas de Computación en el I.S.E.A. (Instituto Superior de Estudios Administrativos), entidad que otorga títulos oficiales reconocidos por el Ministerio de Educación y las prácticas las efectuamos en las MSX.
Me gustaría saber si en los concursos de programas organizados por ustedes, en la sección juegos, sólo deben presentarse programas sobre truco y de inteligencia, o también pueden enviarse algunos referentes a otros juegos como dados.
Kay Carlos A. Fraiese - Mar del Plata

LOAD MSX

Es un placer para nosotros recibir elogios como los tuyos porque nos dan fuerzas para seguir mejorando.
Primero te pedimos disculpas por no haber contestado tu carta anteriormente
Sucede que las cartas se responden respetando el orden de llegada, pero queremos contestarlas todas.
Principalmente el juego debería estimular a pensar y a tomar decisiones como formular preguntas, por ejemplo. Están excluídas las luchas espaciales donde sólo se debe destruir al enemigo y donde el jugador sólo emplea sus reflejos.

Títulos no oficiales

A la computadora personal Talent MSX nada le es imposible

dialogo - D.P.



MSX es marca registrada de MICROSOFT CORPORATION

Porque gracias a la norma internacional MSX, la TALENT MSX trasciende sus propios límites.
Hasta ahora, cuando usted compraba una computadora personal de cualquier marca, quedaba automáticamente desconectado del resto del mundo de la computación. Porque los distintos equipos y sistemas no eran compatibles entre sí.
Hasta que dos grandes empresas de informática, la Microsoft Corp. de EE.UU. y la ASCII del Japón se pusieron de acuerdo para crear una norma standard: la MSX. Que se expandió también rápidamente en Europa. Y que hoy TALENT presenta por primera vez en la Argentina.
Mientras que la mayoría de las computadoras de su tipo que se ofrecen en el mercado nacional, han sido discontinuadas por obsoletas en sus lugares de origen, TALENT MSX tiene casi ilimitadas posibilidades de desarrollo. Porque la norma MSX es en todo el mundo inteligencia en crecimiento.
La TALENT MSX pone a su disposición un mundo de software para elegir. Y con la incorporación de todos sus periféricos llega a ser una auténtica computadora profesional.

UTIL

Su poderoso sistema operativo MSX permite el acceso a todo tipo de procesamiento de datos:

- Planillas de cálculo.
- Procesadores de palabra.
- Gráficos de negocios.
- Bases de datos (d Base II, etc.)
- Contabilidad general, sueldos, y jornales, costos, etc., desarrollados bajo CP/M en Basic, Cobol, Pascal o C.

Con la posibilidad de conexión a línea telefónica permite la transferencia y consulta de datos entre computadoras personales, profesionales o bancos de datos.
La grabación de archivos es en formato MS-DOS, haciéndola compatible con las computadoras profesionales.

DIDACTICA

Dispone de tres lenguajes para la enseñanza de computación: LOGO como lenguaje de inducción para los más chicos. Lenguaje de Programación en castellano, para todos los que quieran aprender a programar sin conocimientos previos. Y Basic MSX como lenguaje profesional. Más una amplia variedad de periféricos como el Mouse, Lapiz Optico, Tableta gráfica, Track-ball, etc.

DIVERTIDA

La más genial para Video-Juegos. Por la amplísima biblioteca de programas **–todos nuevos–** de la norma MSX en el mundo. Y ademas, el Basic MSX permite al usuario generar sus propios juegos con un manejo tan simple, como sólo TALENT MSX puede ofrecer.

CARACTERISTICAS TECNICAS

- Memoria principal 64 KB ampliable hasta 576 KB.
- Memoria de video: 16 KB RAM.
- ROM incorporada de 32 KB con el MSX-Basic de Microsoft.
- Gráficos completos, hasta 32 sprites y 16 colores simultáneos.
- Generador de sonido de 3 voces y 8 octavas.
- Conexión para cualquier grabador.
- Interfaz para salida impresora paralela.
- Conectores para cartuchos y expansiones.
- Fuente para 220 V y modulador PAL-N incorporado.

DISTRIBUIDORES AUTORIZADOS: CAPITAL FEDERAL: AMATRIX, Bolívar 173 - ARGECINT, Av. de Mayo 1402 - BAIDAT COMPUTACION, Juramento 2349 - COMPUPRANDO, Av. de Mayo 965 - COMPUSHOP, Córdoba 1464 - COMPUTIQUE, Córdoba 1111, E. P. - COMPUTRONIC, Viamonte 2096 - CP67 CLUB, Florida 683, L. 18 - DALTON COMPUTACION, Cabildo 2283 - ELAB, Cabildo 730 - MICROSTAR, Callao 462 - Q.S.P., Bartolomé Mitre 864 - SERVICIOS EN INFORMATICA, Paraná 164 - DISTRIBUIDORA CONCALES, Tucumán 1458 - MICROMATICA, Av. Pueyrredón 1135 - **ACASSUSO:** MICROSTAR ACASSUSO, Eduardo Costa 892 - **AVELLANEDA:** ARGOS, Av. Mitre 1755 - **BOULOGNE:** COMPUTIQUE CARREFOUR, Bernardo de Irigoyen 2647 - **CASTELAR:** HOT BIT COMPUTACION, Carlos Casares 997 - **LANUS:** COMPUTACION LANUS, Caaguazú 2186 - **LOMAS DE ZAMORA:** ARGESIS COMPUTACION, Av. Meeks 269 - **MARTINEZ:** VIDEO BYTE, Hipólito Yrigoyen 32 - **RAMOS MEJIA:** MANIAC COMPUTACION, Rivadavia 13734 - **SAN ISIDRO:** FERNANDO CORATELLA, Cosme Beccar 249 - **VICENTE LOPEZ:** SERVICIOS EN INFORMATICA, Av. del Libertador 882 - **BAHIA BLANCA:** SERCOM, Donado 327 - SUMASUR, Alsina 236 - **LA PLATA:** CADEMA, Calle 7 N° 1240 - CERO-UNO INFORMATICA, Calle 48 N° 529 - **MAR DEL PLATA:** FAST, Catamarca 1755 - **NECOCHEA:** CAFAL, Calle 57 N° 2920 - SERCOM, Calle 57 N° 2216 - **TRENQUE LAUQUEN:** COMPUQUEN, Villegas 231 - **CORDOBA:** AUTODATA, Pasaje Santa Catalina 27 - TECSIEM, Santa Rosa 715 - **ROSARIO:** 2001 COMPUTACION, Santa Fe 1468 - MINICOMP, Maipú 862 - SISOR, Urquiza 1062 - **SANTA FE:** ARGECINT, P. San Martín 2433, L. 36 -SISOR, Rivadavia 2553- INFORMATICA, San Gerónimo 2721/25 - **VILLA MARIA:** JUAN CARLOS TRENTO, 9 de Julio 80 - **MENDOZA:** INTERFACE, Sarmiento 98 - BIT & BYTE, 9 de Julio 1030 - **COMODORO RIVADAVIA:** COMPUSER, 25 de Mayo 827 - **GENERAL ROCA:** DISTRIBUIDORA VECCHI, 25 de Mayo 762 - **LA PAMPA:** MARINELLI, Pellegrini 155 - **NEUQUEN:** MEGA, Perito Moreno 383 - EDISA, Roca esq. Fotheringham - **RIO GRANDE:** INFORMATICA M & B, Perito Moreno 290 - **SAN CARLOS DE BARILOCHE:** L. ROBLEDO & ASOCIADOS, Elfein 13, Piso 1° - **TRELEW:** SISTENOVA, Sarmiento 456 -